NASA TECHNICAL NOTE

NASA TN D-6770

c. 1

NASA TN D-6770

AN IMPROVED MULTIPLE LINEAR REGRESSION AND DATA ANALYSIS COMPUTER PROGRAM PACKAGE

by Steven M. Sidik

Lewis Research Center
Cleveland, Ohio 44135

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • APRIL 1972

1. Report No. NASA TN D-6770	2. Government Accession No.
3. Recipient's Catalog No.	
4. Title and Subtitle AN IMPROVED MULTIPLE LINEAR REGRESSION AND DATA ANALYSIS COMPUTER PROGRAM PACKAGE	5. Report Date April 1972
	6. Performing Organization Code
7. Author(s) Steven M. Sidik	8. Performing Organization Report No. E-6718
	10. Work Unit No. 132-80
9. Performing Organization Name and Address Lewis Research Center National Aeronautics and Space Administration Cleveland, Ohio 44135	11. Contract or Grant No.
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546	13. Type of Report and Period Covered Technical Note
	14. Sponsoring Agency Code
15. Supplementary Notes	

16. Abstract

NEWRAP, an improved version of a previous multiple linear regression program called RAPIER, CREDUC, and CRSPLT, allows for a complete regression analysis including cross plots of the independent and dependent variables, correlation coefficients, regression coefficients, analysis of variance tables, t-statistics and their probability levels, rejection of independent variables, plots of residuals against the independent and dependent variables, and a canonical reduction of quadratic response functions useful in optimum seeking experimentation. A major improvement over RAPIER is that all regression calculations are done in double precision arithmetic.

17. Key Words (Suggested by Author(s)) Regression analysis; Multiple linear regression; Least squares; Curve fitting; Response surface analysis	18. Distribution Statement Unclassified - unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	96	$3.00

* For sale by the National Technical Information Service, Springfield, Virginia 22151

CONTENTS

AN IMPROVED MULTIPLE LINEAR REGRESSION AND DATA ANALYSIS COMPUTER PROGRAM PACKAGE

by Steven M. Sidik

Lewis Research Center

SUMMARY

NEWRAP is a digital computer program which can be used with ease to perform extensive regression analyses or a simple least-squares curve fit. The program is written in FORTRAN IV, version 13, for the IBM 7094/7044 DCS. The major value of the program is the comprehensiveness of its calculations and options.

NEWRAP computes the variance-covariance matrix of the independent variables, regression coefficients, t-statistics for individual tests, and analysis of variance tables for overall testing of regression. There is a provision for a choice of three strategies for the variance estimate to be used in computing t-statistics.

Also, more than one set of responses of dependent variables can be analyzed for the same set of independent variables.

A backward rejection option method based on the first dependent variable may be used to delete nonsignificant terms from the model. In this case, a critical significance level is supplied as input. The least significant independent variable is deleted and the regression recomputed. This process is repeated until all remaining variables have significantly nonzero coefficients.

The NEWRAP program uses the triangular form of symmetric matrices throughout. It also allows for the use of weighted regression, computation of predicted values at any combination of independent variables, a table of residuals, and plots of residuals.

By use of CRSPLT, a preregression analysis may be performed which may aid in the choice of model to use in NEWRAP. This program accepts the same raw data in the same format and computes the variance-covariance matrix and correlation matrix of all the variables and an eigenvector decomposition of the variance-covariance matrix corresponding to the independent variables. Microfilm plots are then printed of specified pairs of variables. Punched output of residuals and predicted values from NEWRAP can also be used for more complicated residual plots than the direct use of the plotting option NEWRAP permits.

When a quadratic response function has been estimated (as for example in optimum-seeking experimentation) CREDUC may be used to obtain all information necessary for a canonical analysis of the function.

The three programs together provide a useful data analysis package that can be applied to a large variety of common research and development situations.

INTRODUCTION

RAPIER (ref. 1) is a very flexible multiple linear regression analysis computer program which has been in frequent use at the NASA Lewis Research Center. It was tested with the data presented in Wampler (ref. 2) and performed quite poorly. This alone was not very disturbing since real data are seldom even nearly as ill-conditioned as that set of data. A second factor, however, is that Wampler's data leads to a 5 by 5 matrix to be inverted whereas RAPIER is designed to handle matrices of up to 60 by 60. With real data it is not uncommon for the matrix to become more ill-conditioned as the dimension increases. Often the user increases the size of the model by adding terms which are functions of the original independent variables (as for example in polynomial models) and this often leads to increased correlations and ill-conditioning. For this reason, RAPIER was modified primarily by rearranging the storage of variables in COMMON blocks and performing all the regression calculations in double precision. This was done without losing any of the capabilities of the original program (in fact adding new options). The resulting version is called NEWRAP.

It may be of interest to some RAPIER users that in a number of sample calculations the major numerical inaccuracies arising in the regression calculations were not involved in the actual inversion of the $X'X$ matrix but in the calculation of the inner products which give

$$\hat{b} = (X'X)^{-1}(X'y)$$

Thus a major improvement might be made by computing inner products in double precision arithmetic and truncating to single precision answers without going to complete double precision arithmetic although the latter alternative would further increase the accuracy. As a matter of fact, the double precision inner product calculation is used in a different least-squares method proposed by Golub (ref. 3) which is reference 19 of Wampler's paper.

It should be pointed out that in estimation problems an alternative to the obvious step of more accurate routines is provided by Hoerl and Kennard (ref. 4). They present a technique called "ridge regression" which uses the method of minimum mean squared error estimation in place of minimum variance unbiased estimation. The ridge regression technique should have some definite appeal to statisticians, because it recognizes the fact that existence of ill-conditioned data indicates a problem which should be accounted for statistically as well as computationally. They do consider the problem of rejecting terms but their methods are not amenable to incorporation in NEWRAP in its present form.

A second reason for modifying the program was the desire to provide plots of the residuals as was strongly recommended in chapter 3 of Draper and Smith (ref. 5). With

the microfilm plotting capabilities provided by CINEMATIC (ref. 6) available at the Lewis Research Center computer facility, this feature was also added to NEWRAP without significantly increasing printed output. CINEMATIC is a very specialized set of routines for the 7094/7044 DCS and 360/67 systems. If microfilm plotting is not available at other computer installations, the subroutines used in plotting may readily be changed to routines which produce line printer plots or CALCOMP plots however.

The RAPIER program used an algorithm for the coefficient calculations that inverted the correlation matrix and then converted this to the $(X'X)^{-1}$ matrix to calculate $\hat{b}$. After inspection of several test cases, it seemed that this method did not improve the accuracy of the calculation of $(X'X)^{-1}$. Thus it was dropped and NEWRAP inverts $X'X$ directly.

As with the RAPIER report, only the statistics and mathematics necessary to explain the program capabilities will be presented along with illustrative input and output listings and listings of the programs.

SYMBOLS

B	matrix
b	vector (column)
b_i	true regression coefficient
$\hat{b}_i$	estimated regression coefficient
b_0	constant term
$b_1, \ldots, b_J$	unknown parameters
C	correlation matrix
C_{ij}	elements of C
D	indicator variable, equal to 0 if no b_0 coefficient is estimated and equal to 1 if b_0 is estimated
E(x)	expected value of x (i.e., mean of x over all possible values of x)
e	vector of observed values minus predicted values
$F_{a,d}$	statistic distributed as variance ratio with a and d degrees of freedom
$f_j(z_1, \ldots, z_K)$	term of regression equation
H_0	statistical hypothesis to be tested

H_1	alternate hypothesis to be accepted if H_0 is judged to be false
J	number of coefficients estimated, excluding b_0
K	number of independent variables observed
k	number of segments or cells in range of possible studentized residuals
LOF	lack of fit
M	total number of independent and dependent variables
MS(source)	mean square due to source, where source is REG, RES, etc.
N	number of observations
NPDEG	pooled degrees of freedom for replication error
$N(\mu, \sigma^2)$	normal distribution with mean μ and variance σ^2
R	number of sets of replicates
REG	regression
REP	replication
RES	residual
r_i	number of replicates in set i
S	diagonal matrix
S_c	sum of squares correction if $D = 1$, and 0 if $D = 0$
SSQ(source)	sum of squares due to source, where source is REG, RES, etc.
s_j	elements of diagonal matrix
TOT	total
t_n	statistic distributed as Student's t with n degrees of freedom
V(x)	variance of x, expected value of $(x - E(x))^2$
W, X	matrices
w, x	vectors (column)
X_s	stationary point of estimated quadratic surface
x(J)	x_J
$\bar{x}_{\cdot j}$	$\frac{1}{N}\sum_{i=1}^{N} x_{ij}$
y	vector (column)
Z_i	studentized residual

$z_1, \ldots, z_K$	variables
ϵ	vector of observation errors
μ_x	mean of x defined as E(x)
$\hat{\mu}$	estimate of μ based on observation of random sample
σ_x^2	variance of x defined as V(x)
$\hat{\sigma}^2$	estimate of σ^2 based on observation of random sample
Superscript:	
'	transpose

ESTIMATION OF BASIC LINEAR MODEL

BASIC LINEAR MODEL

In multiple linear regression, a dependent or response variable Y (such as temperature or pressure) measured on an object or experiment is assumed to be correlated with a function of one or more other variables ($z_1, \ldots, z_K$) measured on the same object or experiment. This function includes a number of unknown parameters ($b_1, \ldots, b_J$) and can be represented as

$$y = h(b_1, \ldots, b_J, z_1, \ldots, z_K) + \epsilon \tag{1}$$

The only restriction imposed on this function is that it be linear in the parameters; that is, the function is of the form

$$y = \sum_{j=1}^{J} b_j f_j(z_1, \ldots, z_K) + \epsilon \tag{2}$$

where $f_j(z_1, \ldots, z_K)$ is a TERM of the regression equation. (A TERM is a quantity which may be a variable or a function of a variable, e.g., T is a TERM and Z, after it is defined as Z = log T, is also a TERM.)

Suppose that there are N observations of the dependent variable. Let the subscript i indicate that the values are associated with the i^{th} observation; in particular, the value of the response variable y_i would depend on the observed values of the variables ($z_{i1}, \ldots, z_{iK}$). Also, let the subscript j denote the j^{th} term in the regression

model so that $x_{ij} = f_j(z_{i1}, \ldots, z_{iK})$ describes the transformations of the $z_{i1}, \ldots, z_{iK}$ to produce the value of x_{ij} for the j^{th} term at the i^{th} observation.

The regression model can now be rewritten as

$$y_i = b_1 x_{i1} + b_2 x_{i2} + \ldots + b_J x_{iJ} + \epsilon_i \qquad i = 1, \ldots, N \tag{3}$$

where ϵ_i denotes the difference between the observed value and the expected value of y_i. For the N observations, it is convenient to write this regression model in matrix notation as $y = Xb + \epsilon$ where

$$\left.\begin{aligned}
y &= \begin{pmatrix} y_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ y_N \end{pmatrix} \\
X &= \begin{pmatrix} x_{11} & \cdots\cdots & x_{1J} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ x_{N1} & & x_{NJ} \end{pmatrix} \\
b &= \begin{pmatrix} b_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ b_J \end{pmatrix} \\
\epsilon &= \begin{pmatrix} \epsilon_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \epsilon_N \end{pmatrix}
\end{aligned}\right\} \tag{4}$$

More often than not, the analyst feels the following model is more appropriate:

$$y_i = b_0 + b_1 x_{i1} + \dots + b_J x_{iJ} + \epsilon_i \qquad i = 1, \dots, N \tag{5}$$

Let $a_0 = b_0 + b_1 \bar{x}_{.1} + \dots + b_J \bar{x}_{.J}$. Then, the result of adding this equation to and subtracting it from equation (5) and rearranging the terms is

$$\begin{aligned} y_i = {} & (b_0 + b_1 \bar{x}_{.1} + \dots + b_J \bar{x}_{.J}) \\ & + b_1(x_{i1} - \bar{x}_{.1}) + \dots + b_J(x_{iJ} - \bar{x}_{.J}) + \epsilon_i \qquad i = 1, \dots, N \end{aligned} \tag{6}$$

If, then, a dummy variable x_0 is introduced such that, for all values of i, $x_{i0} = 1.0$, equation (6) may be written as

$$y_i = a_0 x_{i0} + b_1(x_{i1} - \bar{x}_{.1}) + \dots + b_J(x_{iJ} - \bar{x}_{.J}) + \epsilon_i \qquad i = 1, \dots, N \tag{6a}$$

Equation (6a) now resembles equation (3) and may be written in matrix notation, similar to equation (4), as $y = Xb + \epsilon$ where now

$$\left.\begin{aligned} y &= \begin{pmatrix} y_1 \\ \cdot \\ \cdot \\ \cdot \\ y_N \end{pmatrix} \\ x &= \begin{pmatrix} 1.0 & x_{11} - \bar{x}_{.1} & \cdots & x_{1J} - \bar{x}_{.J} \\ 1.0 & x_{21} - \bar{x}_{.1} & \cdots & x_{2J} - \bar{x}_{.J} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ 1.0 & x_{N1} - \bar{x}_{.1} & \cdots & x_{NJ} - \bar{x}_{.J} \end{pmatrix} \\ b &= \begin{pmatrix} a_0 \\ b_1 \\ \cdot \\ \cdot \\ \cdot \\ b_J \end{pmatrix} \\ \epsilon &= \begin{pmatrix} \epsilon_1 \\ \cdot \\ \cdot \\ \cdot \\ \epsilon_N \end{pmatrix} \end{aligned}\right\} \tag{7}$$

ESTIMATING b

Equations (4) and (7) are similar in form and for $N > J$ are an overdetermined set of linear equations. There will be some vector $\hat{b}$ which is a "best" vector to use. If the vector ϵ is composed of random variables ϵ_i such that $E(\epsilon_i) = 0$, $V(\epsilon_i) = \sigma^2 < +\infty$, and the ϵ_i are uncorrelated, then as is well known, the method of least squares gives the linear minimum variance unbiased estimators $\hat{b}$ for b. And $\hat{b}$ is given by

$$\hat{b} = (X'X)^{-1}X'y \tag{8}$$

The matrix $X'X$ divided by $N - 1$ is called the moment matrix of the experiment. The variance-covariance matrix of $\hat{b}$ is given by

$$V(\hat{b}) = \sigma^2(X'X)^{-1} \tag{9}$$

It is important to note that when the form of equation (7) is used, $X'X$ is given by

$$X'X = \begin{pmatrix} N & 0 & \cdots & 0 \\ 0 & \sum_1^N (x_{i1} - \bar{x}_{.1})^2 & \cdots & \sum_1^N (x_{i1} - \bar{x}_{.1})(x_{iJ} - \bar{x}_{.J}) \\ \vdots & \vdots & & \vdots \\ 0 & \sum_1^N (x_{i1} - \bar{x}_{.1})(x_{iJ} - \bar{x}_{.J}) & \cdots & \sum_1^N (x_{iJ} - \bar{x}_{.J})^2 \end{pmatrix} \tag{10}$$

This is seen to be symmetric and of the form

$$X'X = \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}$$

Hence,

$$(X'X)^{-1} = \begin{pmatrix} A^{-1} & 0 \\ 0 & B^{-1} \end{pmatrix}$$

NEWRAP uses this relation to advantage by storing only the lower triangular part of B and computing only the coefficients $b_1, \ldots, b_J$ by matrix manipulations. Then b_0 is given by the simple equation

$$b_0 = \bar{y} - \hat{b}_1\bar{x}_{.1} - \hat{b}_2\bar{x}_{.2} - \cdots - \hat{b}_J\bar{x}_{.J} \tag{11}$$

where $\bar{y} = \sum y_i / N = \hat{a}_0$. It can also be shown that

$$V(\hat{b}_0) = V(\bar{y}) + V(\hat{b}'\bar{x}) = \left[\frac{1}{N} + \bar{x}'(X'X)^{-1}\,\bar{x}\right]\sigma^2$$

$$COV(\hat{b}_0, \hat{b}) = -(X'X)^{-1}\,\bar{x}\sigma^2$$

When there is no b_0 term in the regression model,

$$X'X = \begin{bmatrix} \sum_1^N x_{i1}^2 & \sum_1^N x_{i1}x_{i2} & \cdots & \sum_1^N x_{i1}x_{iJ} \\ \vdots & \vdots & & \vdots \\ \sum_1^N x_{i2}x_{iJ} & \sum_1^N x_{i2}x_{iJ} & \cdots & \sum_1^N x_{iJ}^2 \end{bmatrix} \tag{12}$$

Comparing this to equation (10) shows this form of $X'X$ to be similar to the lower right submatrix in equation (10). This similarity is used to simplify notation by assuming that $X'X$ represents either the form of equation (12) or the lower right portion of equation (10) and considering the calculation of b_0 as a special case. Thus, further reference to b implies

$$b = \begin{pmatrix} b_1 \\ \cdot \\ \cdot \\ \cdot \\ b_J \end{pmatrix}$$

There are two different methods of computing the regression coefficients which may

be used in NEWRAP. The first method uses bordering (ref. 7) on the full X'X matrix. If X'X is a nearly singular matrix, there may be problems with accuracy resulting in overflows or underflows causing execution to terminate without any results being printed. The second method uses a method of bordering which enters one term at a time into the model equation. After each term is entered, a full regression analysis is printed. Typically, if X'X is nearly singular, a number of terms will have been added to the model before the results become unreliable or cause execution to be terminated. Thus, at least a partial analysis of the full model is available to aid in selection of further models to submit. After all the terms have been entered, the program then switches to the procedure which inverts the appropriate full X'X matrix at each stage for further analyses.

The use of the bordering method leads to a large volume of printed output and is not recommended as a standard procedure. Through use of CRSPLT as a preregression analysis program it may be easier to determine if bordering should be used. CRSPLT can also help indicate the order of arrangement of the terms of the model so that those thought to be most important can be entered into the model first.

Also note that the individual observations may be weighted to perform a weighted regression analysis. NEWRAP permits the use of weights (ref. 5). In this case, the X'X and X'y matrices take the following form:

$$\left.\begin{aligned} X'X &= \begin{pmatrix} \sum_{i=1}^{N}\left[(x_{i1} - \bar{x}_{.1})^2 w_i\right] & \cdot \quad \cdot \quad \cdot & \sum_{i=1}^{N}\left[w_i(x_{i1} - \bar{x}_{.1})(x_{iJ} - \bar{x}_{.J})\right] \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \sum_{i=1}^{N}\left[(x_{iJ} - \bar{x}_{.J})(x_{i1} - \bar{x}_{.1})w_i\right] & \cdot \quad \cdot \quad \cdot & \sum_{i=1}^{N}\left[(x_{iJ} - \bar{x}_{.J})^2 w_i\right] \end{pmatrix} \\ X'y &= \begin{pmatrix} \sum_{i=1}^{N} x_{i1}y_iw_i \\ \cdot \\ \cdot \\ \cdot \\ \sum_{i=1}^{N} x_{iJ}y_iw_i \end{pmatrix} \end{aligned}\right\} \qquad (12a)$$

CORRELATION MATRIX

Another matrix of interest both computationally and statistically in the correlation matrix C. The elements of C, which are denoted C_{ij}, are the sample correlation coefficients between the terms X_i and X_j. These are

$$C_{ij} = \sum_{l=1}^{N} \frac{(x_{li} - \bar{x}_{.i})(x_{lj} - \bar{x}_{.j})}{\sqrt{\sum_{k=1}^{N} (x_{ki} - \bar{x}_{.i})^2 \sum_{k=1}^{N} (x_{kj} - \bar{x}_{.j})^2}} \tag{13}$$

and all these numbers are between 1.0 and -1.0.

The calculation of C can be expressed in matrix notation conveniently by defining a diagonal matrix $S = \mathrm{diag}(s_1, s_2, \ldots, s_J)$ with elements

$$s_j = \frac{1.0}{\sqrt{(X'X)_{jj}}} \qquad j = 1, \ldots, J \tag{14}$$

Then

$$C = S(X'X)S \tag{15}$$

It may also be that the independent variables are random variables. Then X'X divided by N - 1 represents the sample variance-covariance matrix and C the sample correlation matrix. If the independent variables are considered to be from a multivariate distribution, it is useful in some cases to consider the eigenvalues and eigenvectors of X'X. For these reasons, NEWRAP includes options to compute and print these quantities. These may also be computed and printed through use of the CRSPLT program.

ESTIMATING σ^2

For any regression model $y = Xb + \epsilon$, there are possibly two methods of estimating σ^2. First, if the assumed regression model is, in reality, the true model, it is well known that an unbiased estimator is given by

$$\hat{\sigma}^2_{RES(J)} = \frac{y'y - \hat{b}'x'y}{N - J - D}$$

$$= \frac{SSQ(RES)}{N - J - D}$$

$$= MS(RES(J)) \tag{16}$$

Second, where there are replicated data points, another estimator of σ^2, depending only on $V(\epsilon_i) = \sigma^2$ for all i and not on the validity of the assumed model, is the pooled mean squares computed from the replicated data points.

Assume the observations are grouped into replicate sets in sequence. Let R be the number of sets of replicates and r_i be the number of replicates in the i^{th} replicate set. Let

$$SSQ(i) = \sum_{k=r^*+1}^{r^*+r_i} (y_k - \bar{y}_i)^2 \tag{17}$$

where

$$r^* = \sum_{j=1}^{i-1} r_j$$

It is assumed y_n is from the i^{th} replicate set and $\bar{y}_i$ is calculated only from those y_n in the i^{th} replicate set. Then define the pooled sum of squares due to replication as $SSQ(REP) = \sum_{i=1}^{R} SSQ(i)$ and the pooled degrees of freedom as $NPDEG = \sum_{i=1}^{R} (r_i - 1)$. The second estimator of σ^2 becomes

$$\sigma^2_{REP} = \frac{SSQ(REP)}{NPDEG}$$

$$= MS(REP) \tag{18}$$

It can be shown (ref. 5, p. 26) that the sums of squares due to residuals can be partitioned into a component due to replication and a component due to lack of fit; that is,

$$SSQ(RES) = SSQ(LOF) + SSQ(REP) \tag{19}$$

This partitioning is used later to determine the estimate of σ^2 to use in tests of hypotheses.

HYPOTHESIS TESTING

NORMALITY OF ϵ

As stated before, the only assumption necessary for $\hat{b}$ to be a linear minimum variance unbiased estimator is that $E(\epsilon_i) = 0.0$, $V(\epsilon_i) = \sigma^2 < +\infty$, and ϵ_i be uncorrelated. If it can further be assumed that $\epsilon_i \sim N(0, \sigma^2)$, a number of standard tests become available. NEWRAP computes a chi-squared statistic which can be used as an approximate test.

Under the hypothesis $\epsilon_i \sim N(0, \sigma^2)$, the studentized residuals defined by

$$Z_i = \frac{y_i - \hat{y}_i}{\hat{\sigma}} = \frac{e_i}{\hat{\sigma}}$$

will be distributed as Student's t with the degrees of freedom associated with the estimate σ. If the degree of freedom is 30 or more, the t distribution is very close to the normal.

The range of possible studentized residuals is $(-\infty, +\infty)$ and may be divided into k segments or cells each with probability p_i, so that each segment will have Np_i as the expected number of observations falling into it. Let n_i denote the number of studentized residuals in the i^{th} cell. Then a chi-squared goodness-of-fit statistic may be calculated as

$$\chi^2_{k-1} = \sum_{i=1}^{k} \frac{(n_i - Np_i)^2}{Np_i}$$

NEWRAP computes this statistic by using an even number of cells greater than or equal to four and less than or equal to 20, such that the expected numbers of observations per cell is five or more. This statistic is not computed when there are less than 20 observations. The bounding values for the i^{th} cell are Z_{i-1}, Z_i where $F(Z_i) = (i \cdot k)/N$ and $F(Z)$ is the cumulative normal distribution function. Then each cell has the same expected number of observations, say $f = N/k$. Then

$$\chi^2_{k-1} = \sum_{i=1}^{k} \frac{(n_i - f)^2}{f} = \frac{k}{N} \sum_{i=1}^{k} n_i^2 - N$$

There is a point to be made concerning the chi-squared calculations. The validity of the use of the chi-squared statistic in a test depends upon the residuals forming a sample of independent and identically distributed random variables. This is not usually the case for regression residuals. Although the tail probabilities of the chi-squared tests might be in error, they should still be able to tell the statistician whether one intended normalizing transformation was more successful than another.

ANALYSIS OF VARIANCE TABLE

For most hypothesis testing of the regression model, it is convenient to summarize the available information in an Analysis of Variance (ANOVA) table, as follows:

Source	Sums of squares	Degrees of freedom	Mean squares
Regression (REG)	$SSQ(REG) = \hat{b}'X'y - S_c$[a]	J	$MS(REG) = SSQ(REG)/J$
Residual (RES)	$SSQ(RES) = y'y - \hat{b}'X'y$	$N - J - D$[b]	$MS(RES) = SSQ(RES)/(N - J - D)$
Total	$SSQ(TOT) = y'y - S_c$	$N - D$	

[a] $S_c = \begin{cases} 0 & \text{if no } b_0 \text{ coefficient is estimated.} \\ N\bar{y}^2 & \text{if a } b_0 \text{ coefficient is estimated.} \end{cases}$

[b] $D = \begin{cases} 0 & \text{if no } b_0 \text{ coefficient is estimated.} \\ 1 & \text{if } b_0 \text{ is estimated.} \end{cases}$

If there are replicated data points, another ANOVA table can be constructed to show the separation of the residual sums of squares into components from lack of fit and replication, as in the following table:

Source	Sums of squares	Degrees of freedom	Mean squares
Lack of fit (LOF)	SSQ(LOF) = SSQ(RES) - SSQ(REP)	N - J - D - NPDEG	MS(LOF) = SSQ(LOF)/(N - J - D - NPDEG)
Replication (REP)	SSQ(REP)	NPDEG	MS(REP) = SSQ(REP)/NPDEG
Residual (RES)	$y'y - \hat{b}'X'y$	N - J - D	

CHOICE OF ESTIMATOR FOR σ^2

As mentioned previously, there are two possible methods of estimating σ^2 depending on whether there are replicated data points. This is true for any given model equation. When the backward rejection option of NEWRAP is used, there is no longer one hypothetical model but a series of models. Thus, there is the choice of estimator for σ^2 to be made after each rejection of a term in the previous model.

As an example, consider the model

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \epsilon \tag{20}$$

with replicated data points. The first step is to estimate b_0, b_1, b_2, and b_3. There will then be the estimators $\hat{\sigma}^2_{RES(J)}$ and $\hat{\sigma}^2_{REP}$. If the model in equation (20) has not left out any important terms, $\hat{\sigma}^2_{RES(J)}$ as well as $\hat{\sigma}^2_{REP}$ is a valid estimator.

The ratio $F = MS(LOF)/MS(REP)$ can be used to test the hypothesis that there is no lack of fit, where $F \sim F_{a,d}$ with $a = N - J - D - NPDEG$ and $d = NPDEG$ degrees of freedom. If the test accepts the hypothesis of no lack of fit, MS(RES) is a pooled estimate of σ^2 with more degrees of freedom. But there is the possibility that the hypothesis was accepted as a result of random fluctuation when there really is some lack of fit; that is, there is the possibility that $\hat{\sigma}^2_{RES(J)}$ is a biased estimator. If lack of fit is not concluded to be significant, the decision to pool or not is usually made on the basis of the number of degrees of freedom for replication. If this is "large" (no definition of large is given herein), $\hat{\sigma}^2_{REP}$ is used. If "small," the pooled estimate $\hat{\sigma}^2_{RES(J)}$ is used.

In testing equation (20), should it be decided that b_3 is not significantly different from zero (see section t-TESTS), the coefficients of the following model would be estimated:

$$y = b_0 + b_1x_1 + b_2x_2 + \epsilon$$

From this model there is an estimate $\hat{\sigma}^2_{RES(J-1)}$. This estimate could also be biased since b_3 may be small but nonzero and the decision of $b_3 = 0$ may have been due to the low power of the test.

At the first step, the lack of fit can be considered a random sample of an infinite possibility of biases. But the biases due to pooling mean squares after rejecting terms can be considered to be systematic biasing. In such a case the use of Cochran's test for "the largest of a set of estimated variances as a fraction of their total" might be appropriate.

NEWRAP provides three strategies of pooling estimates for use in the decision procedure:

(1) Never pool. This is usable only when there are replicated data points. The estimator used in all t-tests is $\hat{\sigma}^2_{REP}$.

(2) Pool initial residual. This will pool the lack of fit and replication (if any) from the first model only. Additional mean squares due to rejected terms will be ignored.

(3) Always pool. This strategy will always use $\hat{\sigma}^2_{RES(J-i)}$ for the model with i rejected terms.

Wherever a $\hat{\sigma}$ or $\hat{\sigma}^2$ is indicated, NEWRAP always uses the value calculated according to the strategy chosen by the user.

TEST OF OVERALL REGRESSION

One of the first tests usually applied to a regression model is the test of the overall significance of the model. In the notation of hypothesis testing this is stated H_0: $b = 0$; H_1: $b \neq 0$ where

$$b = \begin{pmatrix} b_1 \\ \cdot \\ \cdot \\ \cdot \\ b_J \end{pmatrix}$$

The statistic for this test is $F = MS(REG)/\hat{\sigma}^2$. The $F \sim F_{a,d}$ with $a = J - D$, and d equals the degrees of freedom associated with $\hat{\sigma}^2$.

Another useful statistic for judging the significance of overall regression is $R^2 = SSQ(REG)/SSQ(TOT)$. The sampling distribution of R does not lend itself to very simple tests except in the case of H_0: $b = 0$. The main value of R^2 is that it is a

number in the range 0 to 1 and 100 R^2 is a measure of the percentage of variation in the y values that is accounted for by the regression model.

t-TESTS

In many cases, the regression model contains terms whose estimated coefficients are "small." This may be an indication that the term does not have a real effect on the dependent variable and that the estimate is nonzero due to random sampling variation. If this is true, it is desirable to delete the term from the regression model. A test statistic for deciding this is

$$t = \frac{\hat{b}_i}{\sqrt{\hat{\sigma}^2(X'X)^{-1}_{ii}}} \tag{21}$$

where $(X'X)^{-1}_{ii}$ denotes the i^{th} diagonal element of the $(X'X)^{-1}$ matrix. The statistic $t \sim t_{N-J-D}$. An equivalent test statistic is

$$F = t^2 = \frac{\hat{b}_i^2}{\hat{\sigma}^2(X'X)^{-1}_{ii}} \tag{22}$$

where $F \sim F_{1, N-J-D}$. This is often referred to (ref. 5) as the partial F-test. The quantity $\hat{b}_i^2 / \left[(X'X)^{-1}_{ii}\right]$ is called the additional sum of squares due to b_i, if x_i were last to enter the equation. NEWRAP computes and prints the t-statistics, the probability associated with the interval (-t, t), and the additional sums of squares for each term.

This particular test is the basis for the rejection option of NEWRAP. The analyst initially chooses which $\hat{\sigma}^2$ estimator to use by the choice of strategy. Then the analyst may choose a confidence level which all coefficients must meet. For example, suppose a confidence level of 0.900 is chosen. The t-statistic is then computed for each coefficient, and the coefficient with minimum $|t|$ is identified. If $\min|t| > t_{N-J-D, 0.950}$, all terms are concluded to be significant at the 0.1 level (or 90.0 percent level of confidence). If $\min|t| < t_{N-J-D, 0.950}$, the term corresponding to the minimum $|t|$ is dropped from the hypothetical model, and the regression is recomputed. This process is repeated until all remaining coefficients are significant at the specified level of probability. This procedure can be overriden by an option which allows certain specified terms of the model to be retained regardless of the significance of the coefficient. Ken-

nedy and Bancroft (ref. 8) present a study indicating the backward deletion method is slightly more efficient than forward selection in special situations.

PREDICTING VALUES FROM ESTIMATED REGRESSION EQUATION

Regression equations are often used to predict an estimated response at some condition of the independent variables. Useful estimates of parameters to know are the variance of the regression equation and the variance of a single further observation at the desired combination of the independent variables.

Let $x' = (x_1, \ldots, x_J)$ denote the vector of independent variables at which a prediction is desired. Let $x^* = x - \bar{x}$. Let $\hat{\sigma}^2_{\mu \cdot x}$ denote the estimated variance of the regression equation at x. Let $\hat{\sigma}^2_{y \cdot x}$ denote the estimated variance of a single further observation at x. Then,

$$\hat{\sigma}^2_{\mu \cdot x} = \hat{\sigma}^2\left[\frac{D}{N} + x^{*\prime}(X'X)^{-1}\, x^*\right] \tag{23}$$

$$\hat{\sigma}^2_{y \cdot x} = \hat{\sigma}^2\left[1.0 + \frac{D}{N} + x^{*\prime}(X'X)^{-1}\, x^*\right] \tag{24}$$

where, as before, $D = 1$ if a b_0 coefficient is estimated and $D = 0$ if a b_0 coefficient is not estimated. The quantity $s = \hat{\sigma}_{RES(J)}$ is called the standard error of estimate and often is used as a simple approximation to $\hat{\sigma}_{y \cdot x}$. This approximation is close if N is very large and $x = \bar{x}$, in which case,

$$\hat{\sigma}^2_{y \cdot \bar{x}} = s^2\left(1.0 + \frac{D}{N}\right) \approx s^2$$

When $x \neq \bar{x}$, this may be a poor approximation. NEWRAP accepts input vectors x and computes $\hat{y} = \hat{b}_0 + \hat{b}_1 x_1 + \ldots + \hat{b}_J x_J$, as well as $\hat{\sigma}^2_{\mu \cdot x}$, $\hat{\sigma}_{\mu \cdot x}$, $\hat{\sigma}^2_{y \cdot x}$, $\hat{\sigma}_{y \cdot x}$, and the standard error of estimate.

NEWRAP PROGRAM

USERS GUIDE TO NEWRAP INPUT

Illustrative Regression Problem Requiring No Transformations

The illustrative example is described in chapter 7 of reference 5. The data is reproduced in table I. Figure 1 presents this data in a sample input form.

The basic model to be fitted is

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4 \tag{25}$$

TABLE I. - OBSERVED VALUES

FOR EXAMPLE PROBLEM

Unit number	x_1	x_2	x_3	x_4	y
1	-75	0	0	-65	1.4
2	175	0	0	150	26.3
7	0	0	-65	150	29.4
8	0	0	165	-65	9.7
9	0	0	0	150	32.9
10	-75	-75	0	150	26.4
11	175	175	0	-65	8.4
14	-75	-75	-65	150	28.4
15	175	175	165	-65	11.5
18	0	0	-65	-65	1.3
19	0	0	165	150	21.4
20	0	-75	-65	-65	.4
21	0	175	165	150	22.9
24	0	0	0	-65	3.7
3	0	-75	0	150	26.5
5	0	-75	0	150	23.4
16	0	-75	0	150	26.5
4	0	175	0	-65	5.8
6	0	175	0	-65	7.4
17	0	175	0	-65	5.8
12	0	-75	-65	150	28.8
22	0	-75	-65	150	26.4
13	0	175	165	-65	11.8
23	0	175	165	-65	11.4

SAMPLE INPUT

FORTRAN STATEMENT

```
13 SAMPLE NEWRAP PROBLEM 2
    DATA IS FROM   DRAPER AND SMITH
                   APPLIED REGRESSION ANALYSIS  (REFERENCE 4 OF NEWRAP REPORT)
                   CHAPTER 7
   INITIAL MODEL EQUATION IS
          Y = B0 +  B1X1 + B2X2  + B3X3 + B4X4  + ERR

          Y = CHAMBER PRESSURE
          X1 = TEMPERATURE OF CYCLE
          X2 = VIBRATION LEVEL
          X3 = DROP(SHOCK)
          X4 = STATIC FIRE
   RESIDUALS ARE BEING REQUESTED TO BE PUNCHED (FOR CRSPLT PROGRAM).
   FOR RESIDUAL PLOTTING ANALYSES
    4    1  4   24
TTTTTFF1T
    0    0
T950
T
   18    1  1  1   1  1  1  1  1  1  1  1  1  1  1   3   3   2   2
05 (12X,5F6.0)
UNIT NO. 1    -75     0     0   -65   1.4
         2    175     0     0   150  26.3
         7      0     0   -65   150  29.4
         8      0     0   165   -65   9.7
```

NASA-C-836 (REV 9-14-59)

SAMPLE INPUT (Concluded)

FORTRAN STATEMENT

```
         9      0     0     0   150  32.9
        10    -75   -75     0   150  26.4
        11    175   175     0   -65   8.4
        14    -75   -75   -65   150  28.4
        15    175   175   165   -65  11.5
        18      0     0   -65   -65   1.3
        19      0     0   165   150  21.4
        20      0   -75   -65   -65   0.4
        21      0   175   165   150  22.9
        24      0     0     0   -65   3.7
         3      0   -75     0   150  26.5
         5      0   -75     0   150  23.4
        16      0   -75     0   150  26.5
         4      0   175     0   -65   5.8
         6      0   175     0   -65   7.4
        17      0   175     0   -65   5.8
        12      0   -75   -65   150  28.8
        22      0   -75   -65   150  26.4
        13      0   175   165   -65  11.8
        23      0   175   165   -65  11.4
F
```

NASA-C-836 (REV 9-14-59)

Figure 1. - Sample input form.

The preceding model requires no transformations of the tabulated data for the dependent or independent variables. Suitable input statements are also given in figure 1.

A subsequent example will illustrate the requirements on the input cards when transformations are involved.

Detailed Description of Input Cards

This section of the report describes the input cards as classified into nine sets according to table II.

TABLE II. - FUNCTIONS OF INPUT SETS

Set number	Name of set	Purpose
1	IDENTIFICATION	Identify and describe problem
2	PROBLEM SIZE	Define problem size
3	LOGIC	Specify general logical controls
4	MODEL	Define model equation
	(a) MODEL SIZE	
	(b) TERMS	
	(c) TRANSFORMATIONS	
	(d) CONSTANTS	
5	REJECTION	Backward rejection controls
6	REPLICATES	Identify replicated data
7	FORMAT	Give data format
8	DATA	Input observed data
9	PREDICTIONS	Predicted values data

The model equation is defined by set 4 of table II. An example illustrating the use of one blank card for input set 4 which can be used for simple linear regressions is presented by figure 1 and table I. A second example illustrating the use of the set of MODEL cards in the presence of prior constants and transformations will be given at the end of this section of the report. A pictorial representation of an input deck is given by figure 2.

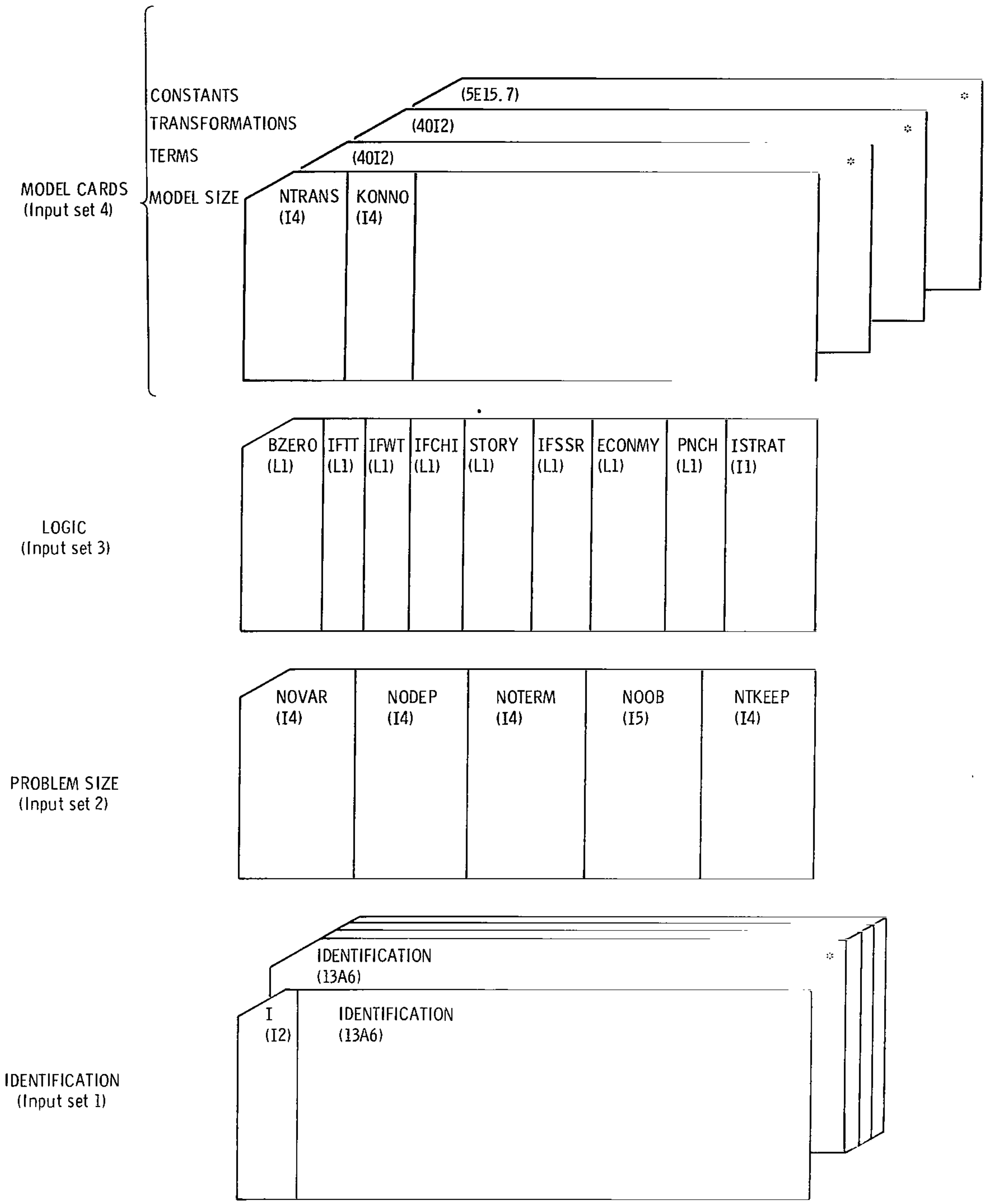

Figure 2. - Sample input deck. (Asterisk denotes the card is optional and its use depends upon data input on previous cards.)

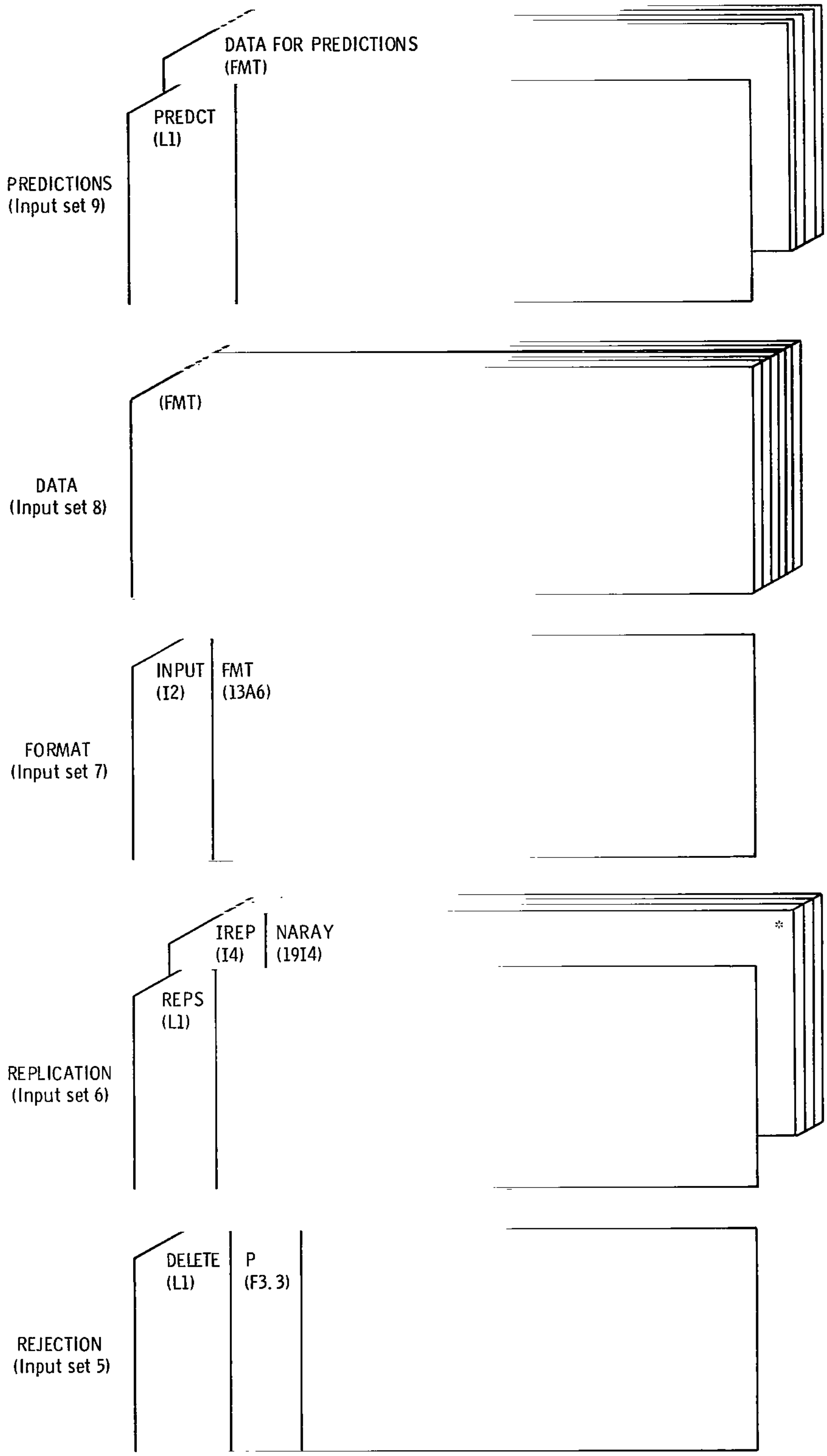

Figure 2. - Concluded.

Details of the input cards are as follows:

(1) IDENTIFICATION (I, IDENT)(I2, 13A6): IDENT is Hollerith data used to identify the problem. I indicates the number of additional cards to read for further identification or description (columns 1 to 78).

(2) PROBLEM SIZE (NOVAR, NODEP, NOTERM, NOOB, NTKEEP)(3I4, I5, I4)

NOVAR	Number of input independent variables (number of z's in eq. (2))
NODEP	Number of input dependent variables
NOTERM	Number of terms in model equation (number of x's in eq. (3)). Note that b_0 is not counted as a term.
NOOB	Number of observations
NTKEEP	First NTKEEP independent terms of model equation will be retained in model regardless of significance level

(3) LOGIC: One card with nine one-column fields

BZERO	b_0 term appears in model equation (T or F)
IFTT	t-statistics and their descriptive confidence levels are to be computed (T or F)
IFWT	Weight of 1.0 is applied to all observations (T or F). If this is F, see input sets 7 and 8 for further information.
IFCHI	Compute and plot residuals (T or F)
STORYX	Calculate eigenvalues and eigenvectors of X'X (T or F)
IFSSR	Model shall be increased by one term at a time using bordering method for matrix inversion (T or F)
ECONMY	Use economy version of output (T or F). NEWRAP does not print X'y, $(X'X)^{-1}$, or C when set to F.
ISTRAT	Pooling strategy is 1, 2, or 3: 1. Never pool. Use replication error as estimate of error. If 1 is selected and no replication is found, strategy 3 is used. 2. Pool initial residual only. 3. Pool all residuals.

PNCH Punch residuals and predicted values (T or F). If T, observation number is punched and then residuals and predicted values are punched in (I6, 4E16.8/(6X, 4E16.8)) format in pairs (observation number, e_1, $\hat{y}_1$, e_2, $\hat{y}_2$, etc.).

(4) MODEL: The MODEL cards are used to manipulate the observed input data, supplied by input set 8, into the form of the desired model equation. There are four subsets of this input set 4, namely, MODEL SIZE, TERMS, TRANSFORMATIONS, and CONSTANTS, of which the latter three are used only in the development of complex models.

If a simple linear model is being analyzed, the MODEL SIZE card is left blank, indicating that the number of transformations is zero and the number of constants to be read in is zero. In this case, the TERMS, TRANSFORMATIONS, and CONSTANTS cards of this input set are not expected by the program, and the program assumes the independent and dependent variables are arranged on the input cards of input set 8 as

$$x_1, x_2, \ldots, x_J, y_1, \ldots, y_{NODEP}$$

where NODEP is the number of dependent variables.

If a weighting factor other than 1.0 is to be used (i.e., if item 3 of the LOGIC card contains an F), the value of the weighting factor for each observation must appear as the last item in the list, so that in this case the data for each observation is entered on the cards as

$$x_1, x_2, \ldots, x_J, y_1, \ldots, y_{NODEP}, WT$$

If the weighting factor is identically 1.0, NEWRAP reads a total of M numerical values for each observation, where M is the sum of the number of independent and dependent variables. The variables are stored consecutively in an array called X, beginning with location 01 and ending with location M. If the weighting factor is not identically 1.0, then M + 1 numerical values are read for each observation, but the last value, being the weighting factor, is treated and stored separately. The data in X are used with their appropriate weighting factors to cumulatively create X'X and X'y as shown in equations (12a).

The remaining discussion of this set explains the use of transformations and/or constants to build more complex models. Therefore, the reader who does not immediately need a complex model may skip this material and proceed directly to the description of input set 5.

As mentioned previously, there are up to four subsets of the MODEL cards. Their purpose is to give the structure of the model equation and thereby specify the initial

operations to be performed on the input data. As used here, CONSTANTS means any numerical value specified to be in the model equation in advance of parameter estimation. These numerical values are read from the CONSTANTS cards.

Also, the word TRANSFORMATIONS is to be interpreted as the operations performed on the input data (read from data cards) to compute the f_j values (eq. (2)) of the model equation. The structure of these functions (and of any transformations of the dependent variables) is read from the TRANSFORMATION cards. Finally, the word TERMS is to be interpreted as the computed results of the operations specified by the transformation (including any operations that leave the input data unchanged). The results of the TRANSFORMATIONS are stored in an array CON, and the TERMS cards designate the order of the relative locations in CON where the final values for the terms of the model equations are to be found.

The four subsets, MODEL SIZE, TERMS, TRANSFORMATIONS, and CONSTANTS, will be described in detail now. Also, at the end of the description of this input set, a summary of these cards, with the formats used, is given for convenience.

The MODEL SIZE card specifies NTRANS and KONNO(2I4) where

NTRANS Number of transformations that will be performed

KONNO Number of constants that will be read in which are required to specify model equation

If the number of transformations is zero, and therefore, the number of constants is zero, the TERMS, TRANSFORMATIONS, and CONSTANTS cards are not expected by the program. This being a simple linear model case, only the MODEL SIZE card, which can be blank, is necessary in this subset, but the values for the observations which are provided in input set 8 must conform to the order as specified in the first three paragraphs describing this input set.

When, however, a more complex model is desired, information must be supplied instructing the program as to (1) where to find the values for the TERMS of the equation, (2) how to create the terms from the variables and the constants, and (3) what the values of the constants are. This information is supplied on the TERMS, TRANSFORMATIONS, and CONSTANTS cards.

The numerical values to be used in the transformations are stored in two arrays called X and CON. The transformations always require that an operator (some value from CON) performs an operation (see table III) on an operand (some specified value from X) to produce a result which will be stored in CON. Thus CON serves two purposes. First, if the number of constants (KONNO) specified on the MODEL SIZE card is nonzero, that many constants will be read from the CONSTANTS cards and stored in CON beginning with location 01. If the number of constants is zero, a CONSTANTS card is not expected by the program. Second, all intermediate and final results of

TABLE III. - OPERATIONS[a] AND CODE NUMBERS

[X indicates a value from X and C a value from CON.]

Operation code (OP)	Resulting operation	Operation code (OP)	Resulting operation
00	No operation	16	1. 0/SQRT(X)
01	N + C	17	C**X
02	X*C	18	10. 0**X
03	C/X	19	SINH(X)
04	EXP(X)	20	COSH(X)
05	X**C	21	(1. 0-COS(X))/2. 0
06	ALOG(X)	22	ATAN(X)
07	ALOG10(X)	23	ATAN2(X, C)
08	SIN(X)	24	X**2
09	COS(X)	25	X**3
10	SIN(π*C*X)	26	ARCSIN(SQRT(X))
11	COS(π*C*X)	27	2. 0*π*X
12	1. 0/X	28	1. 0/(2. 0*π*X)
13	EXP(C/X)	29	ERF(X)
14	EXP(C/X**2)	30	GAMMA(X)
15	SQRT(X)	31	X C

[a]All function names and operations are consistent with FORTRAN IV mathematical subroutines.

transformations are also stored in CON as specified on the TRANSFORMATION cards. The TERMS card then specifies which of the locations in CON finally contain the values needed to construct the X'X and X'y matrices. After all the transformations have been performed on an observation, the contents of the relative locations of the CON array specified on the TERMS card are moved back to X in consecutive locations beginning with location 01.

Note especially that CONSTANTS data are stored in CON from location 01 through KONNO. Thus, if a transformation specifies that a result is to be placed in any of these locations, the result will replace the constant, so that further operations on subsequent transformations would use the new value stored instead of the constant value to which it was initialized. Care should be taken, therefore, that the results of the transformations be stored in relative locations greater than KONNO.

Each transformation code is made up of four subfields of two card columns each, with the following interpretation:

Subfield	Interpretation	
1	Operand	Relative location in X
2	Operation (OP)	Arithmetic operation
3	Operator	Relative location in CON
4	Result	Relative location in CON

Thus, subfield 1 always references the X array, and subfields 3 and 4 reference the CON array. The result of every transformation is a term which is stored in the designated location of the CON array, with the added feature that, if the term is stored in relative location 61 or beyond, it is also stored in the parallel location in the X array. This is illustrated by the arrows in figure 3. This feature allows successive transformations to be performed more easily.

The OP (operation codes) are tabulated in table III. The transformation with OP = 00 is simply an identity transformation. This transfers data from X to CON so that when terms are selected there is a value available in CON that can be moved back to X.

When there are no transformations, NEWRAP assumes the first NOTERM values on a data card are the independent variables and the last NODEP values are the depend-

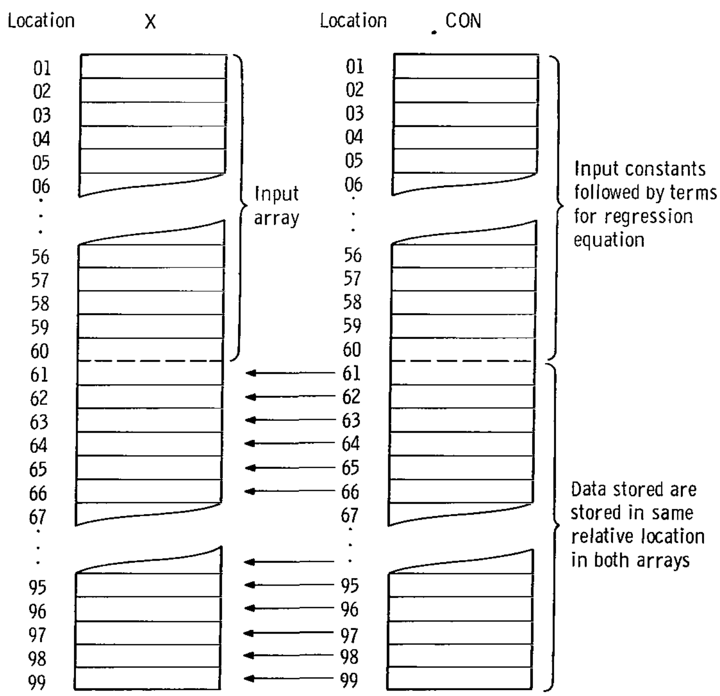

Figure 3. - Map of X and CON arrays. Data transferred into any location of CON array beyond location 60 are immediately duplicated in same relative location in X array.

ent variables. When transformations are used, this convention need not hold for the raw input data but instead holds for the terms on the TERMS card. Thus, the first NOTERM values input on the TERMS card indicate the locations of the CON array which correspond to the independent variables and the last NODEP values indicate the locations of the CON array which correspond to the dependent variables. If the analyst desires to force certain terms to remain in the model regardless of their significance, these terms must be the first terms of the model. Then if the input for NTKEEP of the PROBLEM SIZE card is not zero, the first NTKEEP terms of the model will be retained.

The complete sequence of MODEL cards and the formats used are summarized as follows:

(a) MODEL SIZE (NTRANS, KONNO)(2I4): NTRANS specifies the number of transformations required and KONNO the number of CONSTANTS involved. NTRANS may not be greater than 100 and KONNO not greater than 60. If NTRANS = 0, the following three sets are skipped and the program goes directly to input set 5.

(b) TERMS (40I2): One or more cards as necessary, using two-column fields to denote the relative locations of the CON array containing the final values for the terms to be used and the order in which they enter into the model equation. The number of terms used is specified on the PROBLEM SIZE card.

(c) TRANSFORMATIONS (40I2): As many cards as necessary containing up to 10 transformation instructions per card. Each transformation instruction is composed of four two-column fields. See table III for the list of available transformations.

(d) CONSTANTS (5E15. 7): As many cards as necessary containing the number of CONSTANTS as specified by KONNO. Up to 60 CONSTANTS may be specified. If KONNO = 0, these cards are not expected by the program.

(5) REJECTION (DELETE, P)(L1, F3. 3): If DELETE is set to T, the backward rejection option is used and the desired level of confidence is given by P. The P value is written without a decimal point so that a 95 percent confidence level is indicated by a 950, a 99. 9 percent level as 999, and so forth.

(6) REPLICATION (REPS)(L1): If REPS is F, the program skips to set 7 and assumes there is no replicated data. If REPS is set to T, then more cards are read in 20I4 format specifying:

IREP in the first field of the first card indicates the number of replicate sets.

NARAY in the second field of the first card and the remaining fields of this and succeeding cards consists of an array containing the number of observations in each of the replicate sets.

Note that it is not safe for the program to assume that all the data points for an experiment with the same levels of the independent variables are true replicates. Thus the user must explicitly specify the truly replicated sets. NEWRAP does check that all

independent terms within a replicate set are the same. If not, the program stops. A nonreplicated data point is considered to be a group of size 1. Note that the data in table I are grouped to clearly indicate the replicated data points.

(7) FORMAT (INPUT, FMT)(I2, 13A6): INPUT specifies the unit number on which the input data is stored; and FMT supplies the format for reading it.

Note that, if a weighting factor other than 1.0 is to be used, its value will be read with each data point, and the format must allow for this.

The example from Draper and Smith (ref. 5) uses a weighting value of 1.0 for all data. The format is (12X, 5F6.0) since there are four independent and one dependent variable to be read. If a weighting value other than 1.0 is used, it must appear with every data point as the last value on the card. In such a case, the format could, for example, be (12X, 5F6.0, F10.3).

(8) DATA: Each observation consisting of the given z's and y's read by the execution of one READ statement. Thus, there will be at least one card for each observation. As mentioned previously, if the transformation option is not used, the program expects the first variables to be the independent variables, in the order in which they enter the model, followed by the dependent variables and then the weighting value if IFWT = .F. Otherwise, if transformations are used, the independent and dependent variables may be entered in any convenient order, because the TERMS card(s) will be needed to specify the order in which the values will enter the model equation. However, if IFWT = .F., the weighting value is still the last value supplied with each observation.

(9) PREDICTIONS (PREDCT)(L1):

(DATA) If, with the program LOGIC (input set 3) card, a computation of residuals is requested by a T in card column 4, then predicted values of the dependent variables are computed for all the input values of the independent variables. In addition to these predicted values, predictions at other values of the independent variables might be desired. In PREDICTIONS input, one card with one column is used to indicate if these other predictions are desired (T or F). If this is F, a new case is started and the new case should start with input set 1 cards. If it is T, the following cards are read: One card with one four-column field specifying the number of predictions desired. This is followed by cards with the values of the independent variables at which predictions are desired. Only the final regression model is used, but the number of independent and dependent variables originally supplied on the PROBLEM SIZE data cards are read. All transformations indicated on the MODEL cards are performed. Then the proper terms are chosen by the program to correspond to the final model. Since the dependent variables are not needed in this part of the program, the numerical values for the dependent variables are dummy values and should be in the appropriate range so that when subroutines required for the transformations (e.g., ALOG, SQRT) use these values, abnormal exits will not occur.

Illustrative Problem Requiring Transformations

As an example of the MODEL cards usage consider the following. Suppose the model we are required to construct is

$$\log_{10}(y + 273.15) = b_0 + b_1 z_1 + b_2 z_2 + b_3 z_3 + b_4 z_1 z_2 + b_5 z_1 z_3 + b_6 z_2 z_3 + b_7 z_1^2 + b_8 z_2^2 + b_9 z_3^2$$

Thus, in terms of equation (2) we have

$$x_1 = z_1$$
$$x_2 = z_2$$
$$x_3 = z_3$$
$$x_4 = z_1 z_2$$
$$x_5 = z_1 z_3$$
$$x_6 = z_2 z_3$$
$$x_7 = z_1^2$$
$$x_8 = z_2^2$$
$$x_9 = z_3^2$$
$$y = \log_{10}(y + 273.15)$$

Table IV shows a sequence of transformations which could be used to construct this model equation. Figure 4 shows how the MODEL cards describing this equation would appear on a FORTRAN data sheet. Figure 5 shows the X and CON array contents both before and after the transformations are performed upon one observation and the X array after the appropriate terms have been selected according to the TERMS card data.

TABLE IV. - SEQUENCY OF TRANSFORMATIONS FOR EXAMPLE

Transformation number	Operand	Operation	Operator	Result	Interpretation
1	01	00	00	11	$x_1 \rightarrow$ CON(11)
2	01	00	00	61	$x_1 \rightarrow$ X(61), CON(61)
3	02	00	00	12	$x_2 \rightarrow$ CON(12)
4	02	00	00	62	$x_2 \rightarrow$ X(62), CON(62)
5	03	00	00	13	$x_3 \rightarrow$ CON(13)
6	03	00	00	63	$x_3 \rightarrow$ X(63), CON(63)
7	61	02	61	17	$x_1^2 \rightarrow$ CON(17)
8	62	02	62	18	$x_2^2 \rightarrow$ CON(18)
9	63	02	63	19	$x_3^2 \rightarrow$ CON(19)
10	61	02	62	14	$x_1 x_2 \rightarrow$ CON(14)
11	61	02	63	15	$x_1 x_3 \rightarrow$ CON(15)
12	62	02	63	16	$x_2 x_3 \rightarrow$ CON(16)
13	04	01	01	98	$y + 273.15 \rightarrow$ X(98), CON(98)
14	98	07	00	20	$\log_{10}(y + 273.15) \rightarrow$ CON(20)

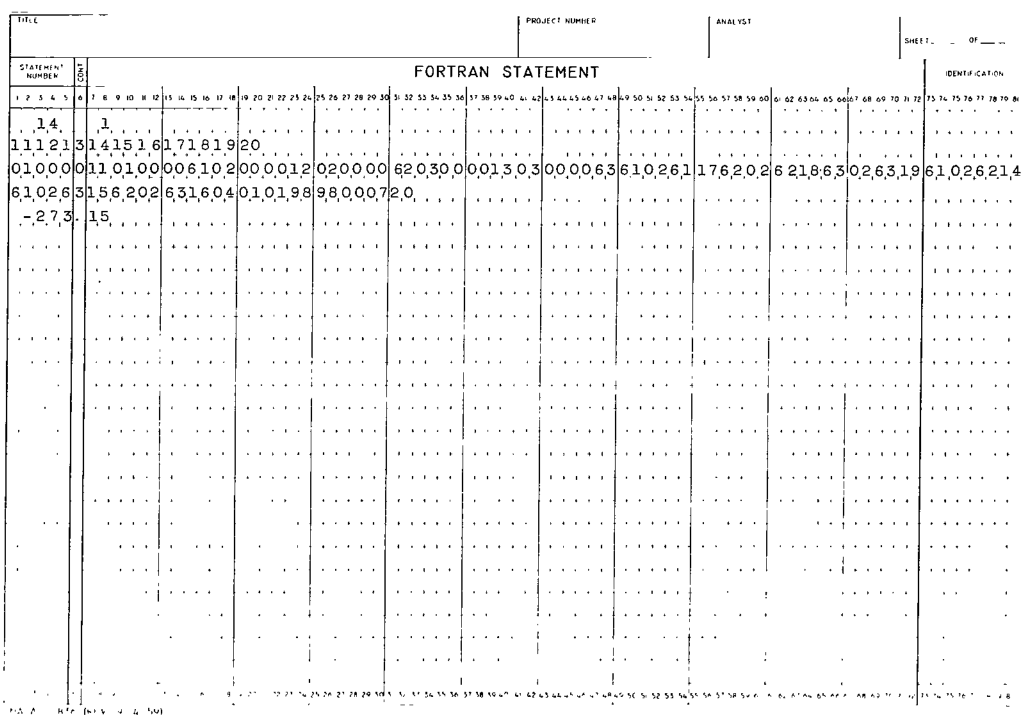

Figure 4. - An example of MODEL cards.

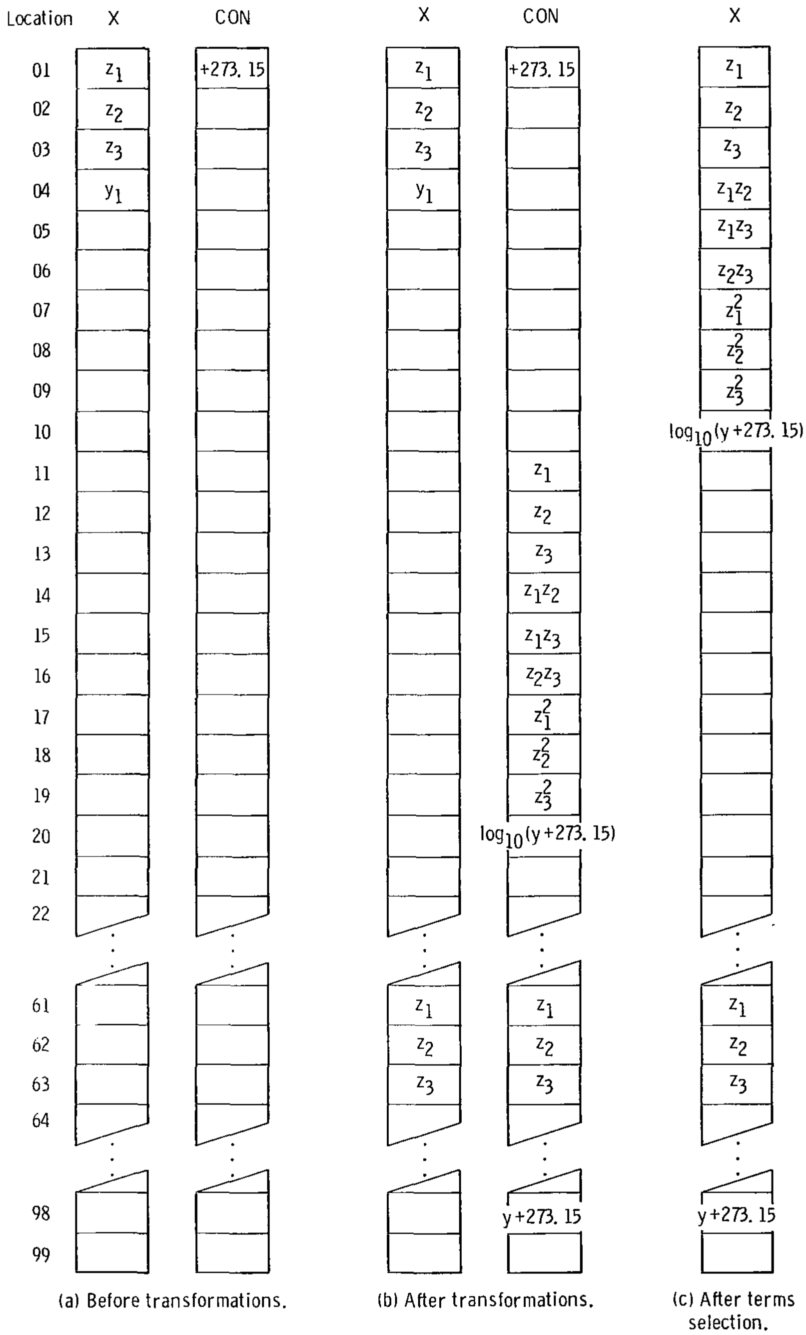

Figure 5. - Arrays X and CON before and after transformations and terms selection for the example.

```
SAMPLE NEWRAP PROBLEM
  DATA IS FROM  DRAPER AND SMITH
                APPLIED REGRESSION ANALYSIS  (REFERENCE 4 OF NEWRAP REPORT)
                CHAPTER 7
  INITIAL MODEL EQUATION IS
       Y= CHAMBER PRESSURE
       X1= TEMPERATURE OF CYCLE
       X2= VIBRATION LEVEL
       X3= DROP(SHOCK)
       X4= STATIC FIRE
       Y =  B0 + B1X1 + B2X2 + B3X3 + B4X4 + ERR
  RESIDUALS ARE BEING REQUESTED TO BE PUNCHED (FOR CRSPLT  PROGRAM)
  FOR RESIDUAL PLOTTING ANALYSES
C*******************************************************************
  4   1   4   24
TTTTTFF1T
THERE IS A B0 TO ESTIMATE
THERE ARE    18  REPLICATE SETS
   1  1  1  1  1  1  1  1  1  1  1  1  1  1  3  3  2  2


(12X,5F6.0)
SAMPLE NEWRAP PROBLEM
TERMS OF THE EQUATION, OBSERVATION =       1
 -75.00000      0              0              -65.00000      1.400000

** REPLICATE SET    1   ************************************************************************************
********************************************************************************************************
TERMS OF THE EQUATION, OBSERVATION =       2
  175.0000      0              0              150.0000       26.30000

** REPLICATE SET    2   ************************************************************************************
********************************************************************************************************
TERMS OF THE EQUATION, OBSERVATION =       3
  0             0              -65.00000      150.0000       29.40000

** REPLICATE SET    3   ************************************************************************************
********************************************************************************************************
TERMS OF THE EQUATION, OBSERVATION =       4
  0             0              165.0000       -65.00000      9.700000

** REPLICATE SET    4   ************************************************************************************
********************************************************************************************************
TERMS OF THE EQUATION, OBSERVATION =       5
  0             0              0              150.0000       32.90000

** REPLICATE SET    5   ************************************************************************************
********************************************************************************************************
TERMS OF THE EQUATION, OBSERVATION =       6
 -75.00000      -75.00000      0              150.0000       26.40000

** REPLICATE SET    6   ************************************************************************************
********************************************************************************************************
TERMS OF THE EQUATION, OBSERVATION =       7
  175.0000      175.0000       0              -65.00000      8.400000

** REPLICATE SET    7   ************************************************************************************
********************************************************************************************************
TERMS OF THE EQUATION, OBSERVATION =       8
 -75.00000      -75.00000      -65.00000      150.0000       28.40000

** REPLICATE SET    8   ************************************************************************************
********************************************************************************************************
TERMS OF THE EQUATION, OBSERVATION =       9
  175.0000      175.0000       165.0000       -65.00000      11.50000

** REPLICATE SET    9   ************************************************************************************
********************************************************************************************************
TERMS OF THE EQUATION, OBSERVATION =      10
  0             0              -65.00000      -65.00000      1.300000

** REPLICATE SET   10   ************************************************************************************
********************************************************************************************************
TERMS OF THE EQUATION, OBSERVATION =      11
  0             0              165.0000       150.0000       21.40000

** REPLICATE SET   11   ************************************************************************************
********************************************************************************************************
TERMS OF THE EQUATION, OBSERVATION =      12
  0             -75.00000      -65.00000      -65.00000      0.400000

** REPLICATE SET   12   ************************************************************************************
********************************************************************************************************
TERMS OF THE EQUATION, OBSERVATION =      13
  0             175.0000       165.0000       150.0000       22.90000

** REPLICATE SET   13   ************************************************************************************
********************************************************************************************************
TERMS OF THE EQUATION, OBSERVATION =      14
  0             0              .              -65.00000      3.700000
```

```
** REPLICATE SET    14  ************************************************************************************************
*********************************************************************************************************************
TERMS OF THE EQUATION, OBSERVATION =     15
 0          -75.00000       0             150.0000        26.50000
TERMS OF THE EQUATION, OBSERVATION =     16
 0          -75.00000                     150.0000        23.40000
TERMS OF THE EQUATION, OBSERVATION =     17
 0          -75.00000       0             150.0000        26.50000

** REPLICATE SET    15  ************************************************************************************************
 DEP. VAR.          1     SSQ= 6.4066836          SUM= 76.400000          MEAN= 25.466666
*********************************************************************************************************************
TERMS OF THE EQUATION, OBSERVATION =     18
 0           175.0000       0             -65.00000       5.800000
TERMS OF THE EQUATION, OBSERVATION =     19
 0           175.0000       0             -65.00000       7.400000
TERMS OF THE EQUATION, OBSERVATION =     20
 0           175.0000                     -65.00000       5.800000

** REPLICATE SET    16  ************************************************************************************************
 DEP. VAR.          1     SSQ= 1.7066677          SUM= 19.000000          MEAN= 5.3333333
*********************************************************************************************************************
TERMS OF THE EQUATION, OBSERVATION =     21
 0          -75.00000     -65.00000       150.0000        28.80000
TERMS OF THE EQUATION, OBSERVATION =     22
 0          -75.00000     -65.00000       150.0000        26.40000

** REPLICATE SET    17  ************************************************************************************************
 DEP. VAR.          1     SSQ= 2.8800044          SUM= 55.200000          MEAN= 27.600000
*********************************************************************************************************************
TERMS OF THE EQUATION, OBSERVATION =     23
 0           175.0000      165.0000       -65.00000       11.80000
TERMS OF THE EQUATION, OBSERVATION =     24
 0           175.0000      165.0000       -65.00000       11.40000

** REPLICATE SET    18  ************************************************************************************************
 DEP. VAR.          1     SSQ= 0.8000137E-01      SUM= 23.200000          MEAN= 11.600000
*********************************************************************************************************************
MEANS OF INDEP AND DEP VARIABLES
  12.500000      33.333333      25.000000      42.500000      16.579167



X TRANSPOSE X MATRIX WHERE X IS DEVIATION FROM MEAN
ROW     1     105000.0
ROW     2     62500.00        263333.3
ROW     3     26250.00        115000.0        173700.0
ROW     4    -26875.00       -161250.0       -49450.00        277350.0



X TRANSPOSE Y MATRIX WHERE X AND Y ARE DEVIATIONS FROM MEAN
ROW     1    -1103.750
ROW     2    -12398.33
ROW     3    -2767.500
ROW     4     25875.25



CORRELATION COEFFICIENTS
ROW     1     1.000000
ROW     2     0.375865        1.000000
ROW     3     0.194372        0.537706        1.000000
ROW     4    -0.157485       -0.596668       -0.225295        1.000000



THE FOLLOWING ARE EIGENVALUES OF X TRANSPOSE X MATRIX
   489039.32        169038.49        99358.063        61947.348



THIS IS THE MODAL MATRIX OR MATRIX OF EIGENVECTORS. EIGENVECTORS ARE WRITTEN IN COLUMNS LEFT TO RIGHT IN SAME ORDER AS EIGENVALUES
ROW     1     0.177713        0.201943        0.820685        0.504097
ROW     2     0.674835        0.220435        0.212133       -0.671570
ROW     3     0.358164        0.666753       -0.500091        0.420793
ROW     4    -0.620269        0.682691        0.177159       -0.343240
```

```
 SAMPLE NEWRAP PROBLEM
EACH COLUMN CONTAINS THE COEFFICIENTS FOR ONE DEPENDENT TERM
CONSTANT TERM (B0)
     11.67123
REGRESSION COEFFICIENTS (B1,...,BK)
  1  0.753117E-02
  2  0.113105E-01
  3  0.428558E-02
  4  0.101374
```

```
ANOVA OF REGRESSION ON DEPENDENT VARIABLE     1
*************************************************************************
     SOURCE            SUMS OF SQUARES     DEGREES OF FREEDOM  MEAN SQUARES
-------------------------------------------------------------------------
    REGRESSION         2462.56296                4            615.640739
    RESIDUAL           249.036608               19            13.107.898
-------------------------------------------------------------------------
    TOTAL              2711.59958               23
*************************************************************************
     R SQUARED = SSQ(REG) / SSQ(TOT) = 0.908159      R = .952974
     STANDARD ERROR OF ESTIMATE     3.620385
USING POOLING STRATEGY  1 THE ERROR MEAN SQUARE =  1.8455595      WITH DEGREES OF FREEDOM =      6
    F=MS(REG)/MS(ERR)= 333.58      COMPARE TO F( 4,  6)

 ANOVA OF LACK OF FIT
*************************************************************************
     SOURCE            SUMS OF SQUARES     DEGREES OF FREEDOM  MEAN SQUARES
-------------------------------------------------------------------------
    LACK OF FIT        237.963249               13            18.3048649
    REPLICATION        11.0733571                6            1.84555951
    RESIDUAL           249.036608               19            13.1071898
-------------------------------------------------------------------------
    F = MS(LOF)/MS(REPS) =       9.918
*************************************************************************
```

```
 SUMS OF SQUARES DUE TO EACH VARIABLE IF IT WERE LAST TO ENTER REGRESSION
  1  5.205730
  2  14.29018
  3  2.221554
  4  1783.654
```

```
 STANDARD DEVIATION OF REGRESSION COEFFICIENTS (DERIVED FROM DIAGONAL ELEMENTS OF  (X TRANSPOSE X)INVERSE MATRIX )
  0  0.354598
  1  0.454375E-02
  2  0.406457E-02
  3  0.390621E-02
  4  0.326098E-02
```

```
(X TRANSPOSE X) INVERSE MATRIX
ROW     1       0.111867E-04
ROW     2      -0.320414E-05   0.895205E-05
ROW     3       0.220215E-06  -0.426580E-05   0.826766E-05
ROW     4      -0.739629E-06   0.413364E-05  -0.984698E-06   0.576159E-05
 SAMPLE NEWRAP PROBLEM

CALCULATED T STATISTICS
THE T STATISTICS CAN BE USED TO TEST THE NET REGRESSION COEFFICIENTS B(I).
  1.679483
  2.782626
  1.097145
  31.08789
UNDER NULL HYPOTHESIS THE INTERVAL (-T,T) WHERE T IS GIVEN ABOVE, HAS APPROX PROBABILITY LISTED BELOW.
 MINUS SIGN INDICATES PROB EXCEEDS .999.
  1          0.856
  2          0.968
  3          0.685
  4         -0.999

THE DESIRED VALUE OF PROBABILITY IS  95.0 PERCENT
          THE TERM X( 3) IS BEING DELETED
```

```
 SAMPLE N-WRAP PROBLEM
EACH COLUMN CONTAINS THE COEFFICIENTS FOR ONE DEPENDENT TERM
CONSTANT TERM (B0)
     11.7 439
REGRESSION COEFFICIENTS (B1,...,BK)
  1  0.751702E-02
  2  0.13 217E-01
  4  0.10.834




ANOVA OF REGRESSION ON DEPENDENT VARIABLE     1
*************************************************************************
      SOURCE          SUMS OF SQUARES     DEGREES OF FREEDOM  MEAN SQUARES
-------------------------------------------------------------------------
    REGRESSION         2460.34143                3             820.113808
    RESIDUAL           251.258162               20             12.5629079
-------------------------------------------------------------------------
    TOTAL              2711.59908               23
*************************************************************************
     R SQUARED = SSQ(REG) / SSQ(TOT) = 0.907339      R = .952544
     STANDARD ERROR OF ESTIMATE     3.544419
USING POOLING STRATEGY  1 THE ERROR MEAN SQUARE =  1.8455595      WITH DEGREES OF FREEDOM =      6
    F=MS(REG)/MS(ERR)= 444.37      COMPARE TO F( 3,  6)

 ANOVA OF LACK OF FIT
*************************************************************************
      SOURCE          SUMS OF SQUARES     DEGREES OF FREEDOM  MEAN SQUARES
-------------------------------------------------------------------------
    LACK OF FIT        240.184803               14             17.1560571
    REPLICATION        11.0733571                6             1.84555951
    RESIDUAL           251.258162               20             12.5629079
-------------------------------------------------------------------------
    F = MS(LOF)/MS(REPS) =       9.296
*************************************************************************




 SUMS OF SQUARES DUE TO EACH VARIABLE IF IT WERE LAST TO ENTER REGRESSION
  1  5.053803
  2  27.08253
  4  1839.096




 STANDARD DEVIATION OF REGRESSION COEFFICIENTS (DERIVED FROM DIAGONAL ELEMENTS OF  (X TRANSPOSE X)INVERSE MATRIX )
  0  0.354384
  1  0.454255E-02
  2  0.352930E-02
  4  0.327752E-02




(X TRANSPOSE X) INVERSE MATRIX
ROW      1      0.111808E-04
ROW      2     -0.309052E-05    0.675106E-05
ROW      3     -0.713401E-06    0.362557E-05    0.564431E-05
 SAMPLE N-WRAP PROBLEM

CALCULATED T STATISTICS
THE T STATISTICS CAN BE USED TO TEST THE NET REGRESSION COEFFICIENTS B(I).
  1.654799
  3.830729
  31.56736
UNDER NULL HYPOTHESIS THE INTERVAL (-T,T) WHERE T IS GIVEN ABOVE, HAS APPROX PROBABILITY LISTED BELOW.
 MINUS SIGN INDICATES PROB EXCEEDS .999.
  1          0.851
  2          0.991
  4         -0.999

THE DESIRED VALUE OF PROBABILITY IS  95.0 PERCENT
          THE TERM X( 1) IS BEING DELETED
```

```
 SAMPLE NEWRAP PROBLEM
EACH COLUMN CONTAINS THE COEFFICIENTS FOR ONE DEPENDENT TERM
CONSTANT TERM (B0)
     11.7 871
REGRESSION COEFFICIENTS (B1,...,BK)
  2  0.155995E-01
  4  0.102354
```

```
ANOVA OF REGRESSION ON DEPENDENT VARIABLE     1
*****************************************************************************
       SOURCE            SUMS OF SQUARES     DEGREES OF FREEDOM  MEAN SQUARES
-----------------------------------------------------------------------------
    REGRESSION          2455.28763                  2              1227.64381
    RESIDUAL            256.311962                 21              12.2053314
-----------------------------------------------------------------------------
    TOTAL               2711.59958                 23
*****************************************************************************
     R SQUARED = SSQ(REG) / SSQ(TOT) = 0.905476      R = .951565
     STANDARD ERROR OF ESTIMATE    3.493613
USING POOLING STRATEGY  1 THE ERROR MEAN SQUARE =  1.8455595     WITH DEGREES OF FREEDOM =      6
    F=MS(REG)/MS(ERR)= 665.19      COMPARE TO F( 2,  6)

 ANOVA OF LACK OF FIT
*****************************************************************************
       SOURCE            SUMS OF SQUARES     DEGREES OF FREEDOM  MEAN SQUARES
-----------------------------------------------------------------------------
    LACK OF FIT         245.238604                 15              16.3492401
    REPLICATION         11.0733571                  6              1.84555951
    RESIDUAL            256.311962                 21              12.2053314
-----------------------------------------------------------------------------
    F = MS(LOF)/MS(REPS) =        8.859
*****************************************************************************
```

```
 SUMS OF SQJARES DUE TO EACH VARIABLE IF IT WERE LAST TO ENTER REGRESSION
  2  41.26720
  4  1871.546
```

```
 STANDARD DEVIATION OF REGRESSION COEFFICIENTS (DERIVED FROM DIAGONAL ELEMENTS OF  (X TRANSPOSE X)INVERSE MATRIX )
  0  0.354375
  2  0.329893E-02
  4  0.321448E-02
```

```
(X TRANSPOSE X) INVERSE MATRIX
ROW     1      0.589681E-05
ROW     2      0.342838E-05    0.559879E-05
 SAMPLE NEWRAP PROBLEM

CALCULATED T STATISTICS
THE T STATISTICS CAN BE USED TO TEST THE NET REGRESSION COEFFICIENTS B(I).
  4.728664
  31.84463
UNDER NULL HYPOTHESIS THE INTERVAL (-T,T) WHERE T IS GIVEN ABOVE, HAS APPROX PROBABILITY LISTED BELOW.
 MINUS SIGN INDICATES PROB EXCEEDS .999.
  2          0.997
  4         -0.999

THE DESIRED VALUE OF PROBABILITY IS  95.0 PERCENT
 SAMPLE NEWRAP PROBLEM
FOR EACH DEPENDENT TERM AND OBSERVATION IS PRINTED
OBSERVED RESPONSE (Y OBSERVED)
CALCULATED RESPONSE (Y CALC)
RESIDUAL (YOBS- YCALC=YDIF)
STUDENTIZED  RESIDUAL ( Z )

Y OBSERVED     1.4000
Y CALC         5.0550
Y DIF         -3.6550
STUDENTIZED   -2.6905

Y OBSERVED     26.300
Y CALC         27.063
Y DIF         -0.7633
STUDENTIZED   -0.5619

Y OBSERVED     29.400
Y CALC         27.063
Y DIF          2.3367
STUDENTIZED    1.7200
```

```
Y OBSERVED     9.7000
Y CALC         5.0550
Y DIF          4.6450
STUDENTIZED    3.4191

Y OBSERVED     32.900
Y CALC         27.063
Y DIF          5.8367
STUDENTIZED    4.2964

Y OBSERVED     26.400
Y CALC         25.893
Y DIF          0.5066
STUDENTIZED    0.3729

Y OBSERVED     8.4000
Y CALC         7.7850
Y DIF          0.6150
STUDENTIZED    0.4527

Y OBSERVED     28.400
Y CALC         25.893
Y DIF          2.5066
STUDENTIZED    1.8451

Y OBSERVED     11.500
Y CALC         7.7850
Y DIF          3.7150
STUDENTIZED    2.7346

Y OBSERVED     1.3000
Y CALC         5.0550
Y DIF         -3.7550
STUDENTIZED   -2.7641
```

SAMPLE NEWRAP PROBLEM

```
Y OBSERVED     21.400
Y CALC         27.063
Y DIF         -5.6633
STUDENTIZED   -4.1688

Y OBSERVED     0.4000
Y CALC         3.8851
Y DIF         -3.4851
STUDENTIZED   -2.5654

Y OBSERVED     22.900
Y CALC         29.793
Y DIF         -6.8932
STUDENTIZED   -5.0741

Y OBSERVED     3.7000
Y CALC         5.0550
Y DIF         -1.3550
STUDENTIZED   -0.9974

Y OBSERVED     26.500
Y CALC         25.893
Y DIF          0.6066
STUDENTIZED    0.4466

Y OBSERVED     23.400
Y CALC         25.893
Y DIF         -2.4934
STUDENTIZED   -1.8354

Y OBSERVED     26.500
Y CALC         25.893
Y DIF          0.6066
STUDENTIZED    0.4466

Y OBSERVED     5.8000
Y CALC         7.7850
Y DIF         -1.9850
STUDENTIZED   -1.4611

Y OBSERVED     7.4000
Y CALC         7.7850
Y DIF         -0.3850
STUDENTIZED   -0.2834

Y OBSERVED     5.8000
Y CALC         7.7850
Y DIF         -1.9850
STUDENTIZED   -1.4611
```

SAMPLE NEWRAP PROBLEM

```
Y OBSERVED     28.800
Y CALC         25.893
Y DIF          2.9066
STUDENTIZED    2.1396
```

```
Y OBSERVED     26.400
Y CALC         25.893
Y DIF          0.5066
STUDENTIZED    0.3729

Y OBSERVED     11.800
Y CALC         7.7850
Y DIF          4.0150
STUDENTIZED    2.9555

Y OBSERVED     11.400
Y CALC         7.7850
Y DIF          3.6150
STUDENTIZED    2.6610



CHI-SQUARE STATISTIC WITH       1 DEGREES OF FREEDOM =     5.000000
SKEWNESS =    8.087309
KURTOSIS =    76.73972
```

```
                                                    HISTOGRAM   1

FREQUENCY     9    2    5    8
  ------------------------------------------------------------------------------------------------------------
     9        *
     8        *              *
     7        *              *
     6        *              *
     5        *         *    *
     4        *         *    *
     3        *         *    *
     2        *    *    *    *
     1        *    *    *    *
  ------------------------------------------------------------------------------------------------------------

INTERVAL      1    2    3    4
CLASS

*01* UNIT05, EOF.                                                              REC=  00000 FIL=  00002
```

NEWRAP DOCUMENTATION AND LISTINGS

The contents of this section include a flow chart of the program, a listing of the routines used in NEWRAP and their major functions, the call structure of the program, a dictionary of the program, and the listing.

General Mathematical and Logical Flow of Program

The flow of operation in NEWRAP is illustrated in figure 6.

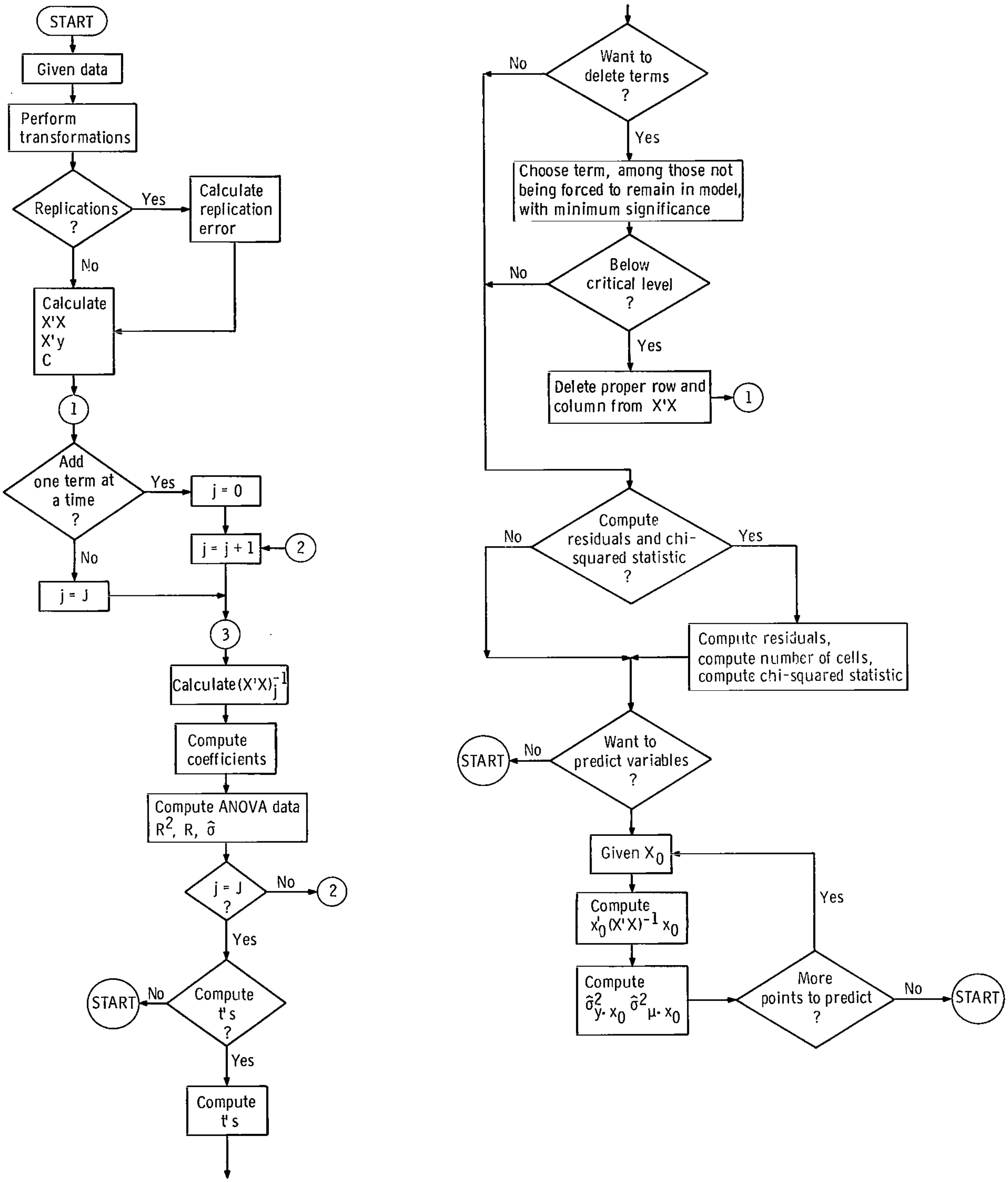

Figure 6. - Flow chart for NEWRAP.

Routines and Their Major Functions

FORTRAN name	Function of routine
BORD	Inverts symmetric matrix of order n by adding bordering column to already inverted matrix of order n - 1
EIGEN	Computes eigenvalues and eigenvectors of input symmetric matrix
HIST	Prints histogram of residuals
INVXTX	Inverts symmetric matrix
LOC	When given row and column indices of symmetric matrix element, it computes location this element would have if only lower triangular part were stored as vector.
MATINV	Controls inversion process; computes regression coefficients; computes eigenvalues and eigenvectors of X'X if requested
MFIX	Prints X'X and computes and prints C
NEWRAP	Executes overall problem control; computes replication error; controls deletion of variables when given results of t-test
OUTPLT	Computes residuals at observed points and plots them. Compute chi-squared statistic
PREDCT	Computes predicted values, variances, and standard deviations of regression line and further observations at specified points
RECT	Writes rectangular matrix
RSTATS	Computes regression statistics and writes regression and lack-of-fit analysis of variance tables
SUMUPS	Constructs X'X and X'y matrices one observation at a time, in double precision
TRAN	Performs transformations
TRIANG	Writes lower triangular part of symmetric matrix
TTEST	Computes t-statistics and their significance levels; determines which variable should be deleted

Call Structure of Program

The call structure of the program is illustrated in figure 7.

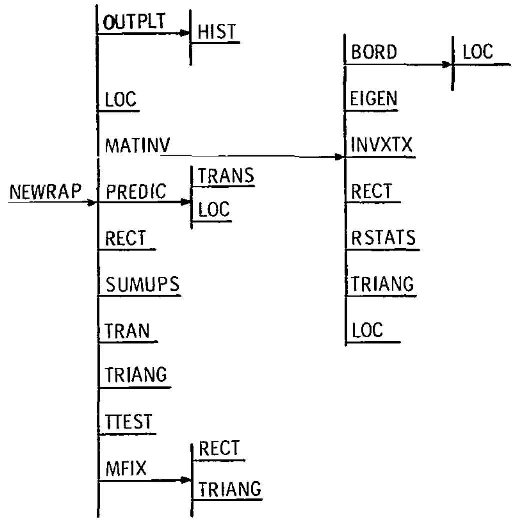

Figure 7. - Call structure of NEWRAP.

Dictionary of Program

FORTRAN name	Mathematical symbol	Description
ACTDEV	e_i	Error in observation i; difference between observed and predicted response
B	b	Regression coefficients other than the constant
BO	b_0	Constant regression coefficient
BZERO		Logical variable set to T if constant b_0 coefficient should be in regression model
CHISQ	χ^2	Chi-squared statistic
CON		Constants used in transformations, and results of transformations
DELETE		Logical variable set to T when deletion of terms is desired
DUMMY		Extra array used for plotting data
ECONMY		Logical variable indicating suppress printout of X'X, X'X deviations, and C if T
ERRMS	$\hat{\sigma}^2$	Estimate of σ^2 used in hypothesis tests
FMT		Variable input format

FMTTRI		Format for printing matrix
IDENT		First identification printed at top of each page
IDOUT		Original sequence number of each term relating reduced models to original model
IFCHI		Logical variable set to T if residual computations and plots are desired
IFSSR		Logical variable set to T if sequential regressions are desired
IFTT		Logical variable set to T if t-statistics are desired
IFWT		Logical variable set to T if all weights of observations are 1.0
INPUT		Input logical tape unit number for data
INPUT5		Set equal to 5 to denote input device is card reader
INTER		Tape unit where input data is stored for use in OUTPLT
IOUT		Sequence number of term among those remaining which is to be deleted
JCOL		Total number of independent and dependent terms in regression model
KONNO		Number of constants originally supplied for transformations
LENGTH		Number of locations in matrix storage area currently needed
LIST		Set equal to 6 to denote output device is printer
NARAY	r_i	Number of replications per replicate set
NCON		Array containing addresses in CON array for use in transformations
NERROR		Degrees of freedom for error mean square estimate
NLOF	N - J - NPDEG - D	Degrees of freedom for estimating variance due to lack of fit
NODEP		Number of dependent variables
NOOB	N	Number of observations

NOTERM	J	Number of terms in current regression model
NOVAR	K	Number of independent variables to be read
NPDEG	NPDEG	Pooled degrees of freedom for replication error
NREG	J	Degrees of freedom for determining variance due to regression
NRES	N - J - D	Degrees of freedom for estimation of residual variance
NTERM		Array containing locations of terms in CON array that should be in regression model
NTOT	N - D	Total degrees of freedom
NTRAN		Array containing transformation codes for use in performing transformations
NTRANS		Number of transformations to perform
NWHERE		Location in X array of first dependent variable; used in prediction routine to adjust for deleted terms
NXCOD		Array containing addresses of variables (or terms with address >60) for use in transformations
P		Probability that interval (-t, t) must have before a term is considered to be significant
PNCH		Logical variable set to T if residuals are to be punched
POOLED	SSQ(REP)	Array containing pooled sums of squares from replications for each dependent term
PREDCT		Logical variable set to T if prediction option is desired
REPS		Logical variable set to T if there are replicate sets in data
REPVAR		Array containing replication variance of each dependent term
RESMS		Array containing residual mean square or variance of each dependent term
RNLOF		Reciprocal of degrees of freedom for lack of fit
RNREG		Reciprocal of degrees of freedom for regression

RNRES		Reciprocal of degrees of freedom for residual
RWT		Reciprocal of total weight
SATRTD		Logical variable indicating that there are no degrees of freedom for residual if T
STORYX		Logical variable set to T if eigenvectors and eigenvalues of X'X are to be computed and printed
SUMX	$\sum x, \sum y$	Array containing sums of independent and dependent terms
SUMX2	$\sum x^2, \sum y^2$	Array containing sum of squared independent and dependent terms
SUMXX	X'X	Sums of squares and crossproducts matrix, and variance-covariance matrix of independent terms
SUMXXI	$(X'X)^{-1}$	Inverse of variance-covariance matrix of independent variables
SUMXY	X'y	Array containing sums of crossproducts of independent terms with dependent terms
TOTWT	w_i	Sum of weight of observations
X		Before transformations are performed, this contains the variables as read in. After transformations are performed, appropriate data from CON array are placed here according to information on TERMS cards.
XCHK		Array used in checking if all values of independent terms are the same within a replicate set
ZEAN	E(X), μ	Expected or mean value (or X)

Program Listing

```
$IBFTC BLDV

      BLOCK DATA
      COMMON /FRMTS/ FMT(13),FMTTRI(14)
      COMMON/BIG/B(60,9),SUMXY(60,9),SUMXX(1830),SUMXXI(1830)
     X,DUMMY(2300)
      DOUBLE PRECISION B,SUMXY,SUMXX,SUMXXI
      COMMON/MED/BO(9),SUMX(69),SUMX2(69),SUMY2(9),ZEAN(69),
     X  CON(99),ERRMS(9),IDENT(13),IDOUT(60),NCON(200),NTERM(69),
```

```
     X  NTRAN(100),NXCOD(100),POOLED(9),REPVAR(9),RESMS(9),X(99)
      DOUBLE PRECISION BO,SUMX,SU MX2,SUMY2,ZEAN
      COMMON/SMALL/     BYPASS,BZERO,DELETE, FIRST, IFCHI,  IFSSR,
     X      IFTT,       IFWT,       INPUT,      INPUT5,     INTER,
     X      ISTRAT,     JCOL,       KONNO,      LENGTH,     LIST,
     X      NERROR,     NODEP,      NOOB,                   NOTERM,
     X      NOVAR,      NPDEG,      NRES,       NTRANS,     NWHERE,
     X      P,          PREDCT,     REPS,       RWT,
     X      STORYI,      STORYC,     STORYX,     TOTWT,      WEIGHT,
     X ERRFXD,  ECONMY,  IOUT,  ICOL
      LOGICAL ECONMY
      DOUBLE PRECISION RWT,TOTWT,WEIGHT
      DATA  INTER/3/,INPUT5/5/, LIST/6/
      DATA (FMTTRI(I),I=1,4)/6H(5H RO, 6HW I5,2, 6HX,(8G1, 6H5.6)) /
      COMMON/MAX/MAXPLT
C     MAXPLT SHOULD BE THE NUMBER OF SINGLE LENGTH WORDS IN COMMON/BIG/
C     BEGINNING AT THE FIRST LOCATION OF SUMXY
      DATA MAXPLT/10700/
      END
```

```
$IBFTC NEWRAP

C
C     THIS IS NEWRAP, MAIN PROGRAM FOR REGRESSION ANALYSIS PROVIDING
C     INTERVAL EVALUATION OF RESULTS.
C**********************************************************************
C
       COMMON /FRMTS/ FMT(13),FMTTRI(14)
      COMMON/BIG/B(60,9),SUMXY(60,9),SUMXX(1830),SUMXXI(1830)
      DOUBLE PRECISION B,SUMXY,SUMXX,SUMXXI
      COMMON/MED/BO(9),SUMX(69),SUMX2(69),SUMY2(9),ZEAN(69),
     X  CON(99),ERRMS(9),IDENT(13),IDOUT(60),NCON(200),NTERM(69),
     X  NTRAN(100),NXCOD(100),POOLED(9),REPVAR(9),RESMS(9),X(99)
      DOUBLE PRECISION BO,SUMX,SU MX2,SUMY2,ZEAN
      COMMON/SMALL/     BYPASS,BZERO,DELETE, FIRST, IFCHI,  IFSSR,
     X      IFTT,       IFWT,       INPUT,      INPUT5,     INTER,
     X      ISTRAT,     JCOL,       KONNO,      LENGTH,     LIST,
     X      NERROR,     NODEP,      NOOB,                   NOTERM,
     X      NOVAR,      NPDEG,      NRES,       NTRANS,     NWHERE,
     X      P,          PREDCT,     REPS,       RWT,
     X      STORYI,      STORYC,     STORYX,     TOTWT,      WEIGHT,
     X ERRFXD,  ECONMY,  IOUT,  ICOL
      LOGICAL ECONMY
      DOUBLE PRECISION RWT,TOTWT,WEIGHT
      LOGICAL     BYPASS,     BZERO,      DELETE,     IFCHI,
     XIFSSR,      IFTT,       IFWT,       REPS,       PREDCT,
     XSTORYC,     STORYX,     STORYI,     FIRST ,ERRFXD
      LOGICAL XSAVE
      LOGICAL PNCH
      DIMENSION XCHK(60)
C
C**********************************************************************
C
      EQUIVALENCE (NARAY,SUMMXI),(S,BO),(SSQ,REPVAR)
      DIMENSION NARAY(1830),S(9),SSQ(9)
C**********************************************************************
C      ZERO OUT ALL DATA ARRAYS EACH NEW DATA SET
  100 DO 101 J=1,4740
```

```
  101 B(J,1)=0.0D0
      DO 102 J=1,225
  102 BO(J)=0.0D0
C
C*********************************************************************
C     READ IDENTIFICATION CARD AND OPTIONS CARD
C
      READ(INPUT5,110) I,IDENT
      WRITE(LIST,111) IDENT
      FIRST=.TRUE.
      ERRFXD=.FALSE.
  113 IF(I) 120,120,115
  115 READ(INPUT5,300) FMT
      WRITE(LIST,301) FMT
      I=I-1
      GO TO 113
  120 READ(INPUT5,1282) NOVAR,NODEP,NOTERM,NOOB,NTKEEP
      WRITE(LIST,1283) NOVAR,NODEP,NOTERM,NOOB
      IF(NTKEEP.NE.0) WRITE(LIST,1307) NTKEEP
      READ(INPUT5,117) BZERO, IFTT, IFWT, IFCHI, STORYX, IFSSR,
     X ECONMY, ISTRAT, PNCH
      WRITE(LIST,118) BZERO, IFTT, IFWT, IFCHI, STORYX, IFSSR, ECONMY,
     XISTRAT, PNCH
C
C*********************************************************************
C     THESE ARE INITIALIZATIONS MADE BEFORE EACH SET OF DATA
C  ICOL DETERMINES THE NUMBER OF VARIABLES READ PER OBSERVATION
C  JCOL IS THE NUMBER OF TERMS IN THE TOTAL REGRESSION EQUATION
C  LENGTH IS THE NUMBER OF STORES NEEDED FOR THE MATRICES
      LENGTH= NOTERM*(NOTERM+1)/2
      ICOL=NOVAR + NODEP
      JCOL = NOTERM +NODEP
      NWHERE= NOTERM
      REWIND INTER
      DO 140  J=1,60
      IDOUT(J) = J
  140 NTERM(J)=J
      DO 145 J=1,100
      NXCOD(J)=J
      NTRAN(J)=0
  145 NCON(2*J)=J
C
C*********************************************************************
      IF(BZERO) WRITE(LIST,190)
      IF(.NOT.BZERO) WRITE(LIST,170)
C*********************************************************************
      READ(INPUT5,282) NTRANS,KONNO
      IF(NTRANS.EQ.0) GO TO 255
  220 READ  (INPUT5,230)(NTERM(K),K=1,JCOL)
      WRITE(LIST,235) (NTERM(K),K=1,JCOL)
      READ  (INPUT5,230)(NXCOD(I),NTRAN(I),NCON(2*I-1),NCON(2*I),I=1,NTR
     AANS )
      WRITE(LIST,240) (NXCOD(I),NTRAN(I),NCON(2*I-1),NCON(2*I),I=1,
     X    NTRANS)
      IF(KONNO) 255,255,250
  250 READ  (INPUT5,260)(CON(I),I=1,KONNO)
      WRITE(LIST,262) (CON(I),I=1,KONNO)
C*********************************************************************
C
  255 READ(INPUT5,257) DELETE,P
      IF(DELETE) IFTT=.TRUE.
C
```

```
C***********************************************************************
C      IF THERE ARE REPLICATED POINTS READ IN THE NUMBER OF POINTS AND
C        THE NUMBER OF REPLICATIONS.  SINGLE DATA POINTS ARE DATA POINTS
C      REPLICATED ONCE.  OBSERVED DATA MUST BE ARRANGED IN THE ORDER
C      IMPLIED HERE.
       READ(INPUT5,257) REPS
       XSAVE=.FALSE.
  265  IF(.NOT.REPS) GO TO 290
       READ(INPUT5,282) IREP,(NARAY(I),I=1,IREP)
       WRITE(LIST,284) IREP
       WRITE(LIST,283)(NARAY(I),I=1,IREP)
       NPDEG=0
       IREP=1
       IC=NARAY(1)
       XSAVE=.TRUE.
       DO 315 I=1,NODEP
       POOLED(I)=0.0
       S(I)=0.0
  315  SSQ(I)=0.0
C
C***********************************************************************
C      READ VARIABLE FORMAT FOR DATA
  290  READ(INPUT5,110) INPUT,FMT
       WRITE(LIST,111) FMT
  310  TOTWT=0.0D0
       WEIGHT=1.0D0
        WRITE(LIST,301) IDENT
C
C***********************************************************************
C       READ IN INPUT VARIABLES
       DO 490 J=1,NOOB
  330  IF(.NOT.IFWT) GO TO 350
  340  READ (INPUT,FMT) (X(I),I=1,ICOL)
       GO TO 360
  350  READ (INPUT,FMT)(X(I),I=1,ICOL), WEIGHT
  360  CONTINUE
       IF(ECONMY) WRITE(LIST,381) J,(X(I),I=1,ICOL)
  381  FORMAT(1H I4,9G14.6/(5X,9G14.6))
       IF(NTRANS.EQ.0) GO TO 450
       IF(ECONMY) GO TO 390
       WRITE(LIST,370)WEIGHT,J
       WRITE (LIST,380)(X(I),I=1,ICOL)
  390  CALL TRANS
  420  DO 430      K=1,JCOL
       I=NTERM(K)
       X(K) = CON(I)
  430  CONTINUE
  450  CONTINUE
       IF(ECONMY) GO TO 4609
       WRITE(LIST,460) J
  461  WRITE (LIST,380)(X(I),I=1,JCOL)
 4609  CONTINUE
       IF(IFCHI) WRITE(INTER) (X(I),I=1,69),WEIGHT
       IF(.NOT.XSAVE) GO TO 4611
       DO 4610 K=1,NOTERM
 4610  XCHK(K)=X(K)
       XSAVE=.FALSE.
 4611  CONTINUE
C
C***********************************************************************
```

```
C       COMPUTE THE ERROR VARIANCE FROM REPLICATED DATA
        IF(.NOT.REPS) GO TO 480
        IGOTO =1
        IF(NARAY(IREP).GT.1) IGOTO=2
        IF(J.GE.IC) WRITE(6,462) IREP
        DO 475 I=1,NODEP
        IF(I-1) 4629,4629,464
 4629   DO 463 K=1,NOTERM
        IF(X(K).NE.XCHK(K)) GO TO 2001
  463   CONTINUE
  464   CONTINUE
        KBAR=NOTERM+I
        S(I)=S(I)+X(KBAR)
        SSQ(I)=SSQ(I)+X(KBAR)**2
        IF(J-IC) 475,465,465
  465   GO TO (468,466),IGOTO
  466   ZEAN(1)=S(I)/FLOAT(NARAY(IREP))
        SSQ(I) = SSQ(I) - ZEAN(1)*S(I)
        POOLED(I)=POOLED(I)+SSQ(I)
        WRITE(LIST,467) I,SSQ(I),S(I),ZEAN(1)
  468   IF(I.LT.NODEP) GO TO 469
        NPDEG=NPDEG+NARAY(IREP) -1
        IREP=IREP+1
        IC = IC + NARAY(IREP)
        WRITE(LIST,4671)
  469   S(I)=0.0
        SSQ(I)=0.0
        XSAVE=.TRUE.
  475   CONTINUE
C
C***********************************************************************
C       CALCULATE SUMS, SUMS OF SQUARES AND SUMS OF CROSS PRODUCTS.
  480   CALL SUMUP
  490   CONTINUE
C       490 CONTINUE IS THE END OF THE LOOP FOR READING DATA CARDS
        IF(.NOT.REPS) GO TO 496
        DO 493 I=1,NODEP
        REPVAR(I)=POOLED(I)/FLOAT(NPDEG)
  493   CONTINUE
  496   CONTINUE
C
C***********************************************************************
C       ALL DATA HAS BEEN READ IN AND THE   XTRANSPOSEX AND XTRANSPOSEY
C       MATRIX HAVE BEEN CALCULATED.
C       NOW WRITE THE MATRICES
        CALL MFIX
        REWIND INTER
        GO TO 640
C
C***********************************************************************
C       THIS CODING DELETES THE DATA FROM THE SUMXX MATRIX
C       CORRESPONDING TO THE INDEPENDENT TERM DELETED
 6500   CONTINUE
        IR=IOUT-1
        IC= NOTERM - IOUT
        IF ( IC.EQ. 0) GO TO 6700
        INOCH= IOUT*IR/2
        INEW = INOCH
        IOLD = INEW + IOUT
        IRC=0
        IBC=0
        ITC=0
```

```
      DO 6600 I=IOLD,LENGTH
      INEW = INEW+1
      IOLD=IOLD + 1
      IF(ITC.GT.0) GO TO 6540
      IRC=IRC + 1
      IF(IRC.GT.IR)  GO TO 6530
      SUMXX(INEW)=SUMXX(IOLD)
      GO TO 6600
 6530 IBC=IBC + 1
      ITC = IBC
      IOLD = IOLD+1
      IRC= 0
 6540 ITC = ITC -1
      SUMXX(INEW)=SUMXX(IOLD)
 6600 CONTINUE
 6700 LENGTH = LENGTH-NOTERM
      NOTERM= NOTERM -1
      JCOL= NOTERM+NODEP
C
C*********************************************************************
C     INVERT THE SUMXX MATRIX AND COMPUTE REGRESSION COEFS
C     AND SUMS OF SQUARES DUE TO REGRESSION IN THE MATRIX INVERSION
C     ROUTINE
  640 CONTINUE
      CALL MATINV
      FIRST=.FALSE.
C
C*********************************************************************
C     WRITE(XTX) INVERSE. THIS MATRIX TIMES ERROR MEAN SQUARE (ERRMS)
C     IS THE VARIANCE-COVARIANCE MATRIX OF REGRESSION COEFFICIENTS.
      IF(ECONMY) GO TO 970
      WRITE(LIST,700)
       CALL TRIANG(X,SUMXXI,NOTERM,8,FMTTRI,2)
C
C*********************************************************************
C     IF A VARIABLE HAS BEEN DELETED ADJUST COUNTERS AND RECOMPUTE THE
C     REGRESSION. IF NO VARIABLE HAS BEEN DELETED CONTROL WILL PASS
C     FROM THE TTEST ROUTINE TO THE CHI-SQUARE OPTION.
  970 CONTINUE
      IF(.NOT.IFTT) GO TO 1020
  980 WRITE (LIST,301)IDENT
      CALL TTEST($1020,NTKEEP)
      IF(NODEP-1) 985,990,985
  985 WRITE(LIST,986) NODEP
      NODEP=1
  990 J=JCOL-1
      DO 995 K=IOUT,J
      NTERM(K)=NTERM(K+1)
      ZEAN(K) = ZEAN(K+1)
      SUMX(K) = SUMX(K+1)
      SUMX2(K) = SUMX2(K+1)
      IDOUT(K) = IDOUT(K+1)
      SUMXY(K,1)= SUMXY(K+1,1)
  995 CONTINUE
      IF(NOTERM.EQ.1) GO TO 1000
      GO TO 6500
 1000 WRITE(LIST,1005)
      NOTERM=0
      GO TO 1035
C
C*********************************************************************
C
```

```
 1020 IF(.NOT.IFCHI) GO TO 1035
 1030 WRITE(LIST,301) IDENT
      CALL OUTPLT(PNCH)
C
C***********************************************************************
 1035 READ(INPUT5,117) PREDCT
      IF(.NOT.PREDCT) GO TO 100
      CALL PREDIC
 1040 GO TO 100
C
C***********************************************************************
 2001 WRITE(LIST,1306)
      STOP
C***********************************************************************
 8001 FORMAT(1H1)
 8002 FORMAT(1H2)
  110 FORMAT (I2,13A6)
  111 FORMAT (1H1,13A6,A2)
  117 FORMAT(7L1,I1,L1)
  118 FORMAT(1H 7L1,I1,L1)
  170 FORMAT(33H THERE IS NO B0 TERM IN THE MODEL)
  190 FORMAT(26H THERE IS A B0 TO ESTIMATE )
  230 FORMAT(40I2)
  235 FORMAT(11H NTERM(K)=  /(1H 30I4))
  240 FORMAT(25H THE TRANSFORMATIONS ARE  /(1H 5(4I4,5X)))
  257 FORMAT( 1L1, F3.3)
  260 FORMAT(5E15.7)
  262 FORMAT(19H THE CONSTANTS ARE  /((1H 8G15.7)))
  282 FORMAT(20I4)
  283 FORMAT(1H 20I4)
  284 FORMAT(11H THERE ARE  I5,16H  REPLICATE SETS )
  300 FORMAT(13A6,1A2)
  301 FORMAT (1H 13A6,A2)
  370 FORMAT(1H0,29HOBSERVED VARIABLES, WEIGHT = G14.6,6X,15HOBSERVATION
     1 = ,I5)
  380 FORMAT(1H  9G14.6)
  460 FORMAT(1H ,37HTERMS OF THE EQUATION, OBSERVATION = ,I5)
  462 FORMAT(18HK** REPLICATE SET  I5,3X,100(1H*))
 4671 FORMAT(1H 125(1H*))
  467 FORMAT(14H  DEP. VAR.    I6,8H    SSQ=G14.7,8H    SUM=G14.7,8H   M
     XEAN= G14.7)
  540 FORMAT(1H 8G14.7)
  560 FORMAT(21H2X TRANSPOSE X MATRIX    )
  670 FORMAT(25H2CORRELATION COEFFICIENTS   )
  700 FORMAT(32H2(X TRANSPOSE X) INVERSE MATRIX      )
  986 FORMAT(39H THE NUMBER OF DEPENDENT VARIABLES WAS  I3,83H  IT IS BE
     XING SET TO ONE AND THE REJECTION OPTION EXERCISED ON DEPENDENT VAR
     XIABLE 1   )
 1005 FORMAT(39H THERE IS NO EVIDENCE OF A REGRESSION.     /
     X     74H USE THE MEAN RESPONSE FOR THE BEST ESTIMATE OF THE DEPEND
     XENT VARIABLE(S).   )
 1282 FORMAT(3I4,I5,I4)
 1283 FORMAT(1H 3I4,I5)
 1306 FORMAT(40H REPLICATE SETS ARE NOT GROUPED PROPERLY    )
 1307 FORMAT(11H THE FIRST  I2,64H TERMS OF THE MODEL WILL BE RETAINED R
     XEGARDLESS OF SIGNIFICANCE    )
      END
```

```
$IBFTC MATINX

C
C     SUBROUTINE MATINV
C
C***********************************************************************
C
C     PURPOSE
C         1)COMPUTE EIGENVALUES AND EIGENVECTORS OF (X-TRANSPOSE X)
C            MATRIX IF REQUESTED. (STORYX=.TRUE.)
C         2)COMPUTE (X TRANSPOSE X) INVERSE
C         3)COMPUTE REGRESSION COEFFICIENTS
C         4)COMPUTE OTHER REGRESSION STATISTICS
C
C     SUBROUTINES NEEDED
C           BORD
C           LOC
C           EIGEN
C           INVXTX
C           RECT
C           RSTATS
C           TRIANG
C
C     REMARKS
C           THE EIGENVALUES ARE COMPUTED AS AN AID IN DETERMINING THE
C           CONDITION OF THE SYSTEM OF EQUATIONS FOR THE REGRESSION
C           COEFFICIENTS. EXAMINATION OF THEM AND THEIR ASSOCIATED
C           EIGENVECTORS MAY SHOW THAT CERTAIN SETS OF INDEPENDENT
C           VARIABLES ARE HIGHLY CORRELATED AND NOT EASILY LIABLE TO
C           INDEPENDENT STUDY.
C
      SUBROUTINE MATINV
C
C***********************************************************************
      COMMON/BIG/B(60,9),SUMXY(60,9),SUMXX(1830),SUMXXI(1830)
      DOUBLE PRECISION B,SUMXY,SUMXX,SUMXXI
      COMMON/MED/BO(9),SUMX(69),SUMX2(69),SUMY2(9),ZEAN(69),
     X  CON(99),ERRMS(9),IDENT(13),IDOUT(60),NCON(200),NTERM(69),
     X  NTRAN(100),NXCOD(100),POOLED(9),REPVAR(9),RESMS(9),X(99)
      DOUBLE PRECISION BO,SUMX,SU MX2,SUMY2,ZEAN
       COMMON /FRMTS/ FMT(13),FMTTRI(14)
      COMMON/SMALL/      BYPASS,BZERO,DELETE, FIRST, IFCHI,   IFSSR,
     X      IFTT,        IFWT,        INPUT,       INPUT5,      INTER,
     X      ISTRAT,      JCOL,        KONNO,       LENGTH,      LIST,
     X      NERROR,      NODEP,       NOOB,                     NOTERM,
     X      NOVAR,       NPDEG,       NRES,        NTRANS,      NWHERE,
     X      P,           PREDCT,      REPS,        RWT,
     X      STORYI,       STORYC,      STORYX,      TOTWT,       WEIGHT,
     X ERRFXD,  ECONMY,  IOUT,  ICOL
      LOGICAL ECONMY
      LOGICAL     BYPASS,     BZERO,      DELETE,     IFCHI,
     XIFSSR,      IFTT,       IFWT,       REPS,       PREDCT,
     XSTORYC,     STORYX,     STORYI,     FIRST ,ERRFXD
      DOUBLE PRECISION RWT,TOTWT,WEIGHT
      DIMENSION A(1),C(1),XTX(3)
      EQUIVALENCE (A,SUMXXI),(C,SUMXXI(915))
      DOUBLE PRECISION SUM
      DATA  (XTX(I),I=1,3) /6HX TRAN, 6HSPOSE , 6HX     /
C
C***********************************************************************
      IORDER= NOTERM
      IF(NOTERM-1) 10,10,12
```

```
   10 SUMXXI(1)= 1.0/SUMX2(1)
      GO TO 350
C
C
   12 IF(.NOT. STORYX) GO TO 30
      DO 14 I=1,LENGTH
      A(I)=SUMXX(I)
   14 CONTINUE
   16 CALL EIGEN(A,C      ,IORDER,0)
      WRITE(LIST,17)(XTX(I),I=1,3)
      J=0
      DO 18 I=1,IORDER
      J=J+I
   18 A(I)=A(J)
      WRITE(LIST,19) (A(I),I=1,IORDER)
      WRITE(LIST,20)
      CALL RECT(IORDER,IORDER,IORDER,IORDER,C,X    ,FMTTRI,1)
   30 DO 35 I=1,LENGTH
   35 SUMXXI(I)=SUMXX(I)
C
C**********************************************************************
C       NO SUBMODELS TO ANALYZE SO INVERT A DIRECTLY BY GAUSS
   49 IF(IFSSR) GO TO 50
      CALL INVXTX(SUMXXI,NOTERM,D,1.0)
      GO TO 60
C
C**********************************************************************
CC        SUBMODELS HAVE BEEN REQUESTED SO WE USE BORDERING
   50 IORDER=0
   55 IORDER=IORDER +1
      CALL BORD(IORDER,SUMXXI)
   60 CONTINUE
C
C**********************************************************************
C     COMPUTE COEFFICIENTS AND PRINT THEM
C
  350 DO 370 J=1,NODEP
      DO 370 K=1,IORDER
      B(K,J)=0.0D0
      DO 370 L=1,IORDER
      CALL LOC(L,K,IR)
      B(K,J) = B(K,J) + SUMXXI(IR)*SUMXY(L,J)
  370 CONTINUE
C
      WRITE(LIST,380) IDENT
      WRITE(LIST,382)
      IF(.NOT.BZERO) GO TO 400
      DO 390 J=1,NODEP
      SUM=0.0D0
      KBAR= NOTERM + J
      DO 385 K=1,IORDER
      SUM = SUM + B(K,J)*ZEAN(K)
  385 CONTINUE
      BO(J)= ZEAN(KBAR) -SUM
  390 CONTINUE
      WRITE(LIST,395)
      WRITE(LIST,397) (BO(K),K=1,NODEP)
  400 WRITE(LIST,410)
      DO 430 J=1,IORDER
      WRITE(LIST,432) IDOUT(J),(B(J,K),K=1,NODEP)
  430 CONTINUE
C
```

```
C*********************************************************************
C      COMPUTE REGRESSION STATISTICS IN RSTATS
C
       CALL RSTATS(IORDER)
C
C*********************************************************************
C      IF IORDER IS LESS THAN NOTERM WE HAVE USED THE BORDERING OPTION
C      AND MUST GO BACK TO FINISH.
C
       IF(IORDER-NOTERM) 55,500,500
  500  STORYX=.FALSE.
       IFSSR=.FALSE.
       RETURN
   17  FORMAT(34H2THE FOLLOWING ARE EIGENVALUES OF  2A6,A1,  7H MATRIX)
   19  FORMAT(1H 8G16.7)
   20  FORMAT(132H2THIS IS THE MODAL MATRIX OR MATRIX OF EIGENVECTORS. EI
      1GENVECTORS ARE WRITTEN IN COLUMNS LEFT TO RIGHT IN SAME ORDER AS
      2EIGENVALUES )
  380  FORMAT(1H1,13A6,1A2)
  382  FORMAT( 61H EACH COLUMN CONTAINS THE COEFFICIENTS FOR ONE DEPENDEN
      XT TERM )
  395  FORMAT(20H CONSTANT TERM (B0) )
  397  FORMAT(4X,9G14.6)
  410  FORMAT(36H REGRESSION COEFFICIENTS (B1,...,BK) )
  432  FORMAT(1H I3,9G14.6)
       END
```

```
$IBFTC TTESTX

C*********************************************************************
C
C      SUBROUTINE TTEST
C
C      PURPOSE
C            COMPUTE THE T-STATISTICS FOR EACH REGRESSION TERM AND
C            ITS TWO TAILED SIGNIFICANCE LEVEL.  THEN DETERMINE THE
C            TERM WITH THE LEAST SIGNIFICANCE AND RETURN THIS
C            INFORMATION TO NEWRAP
C
C*********************************************************************
       SUBROUTINE TTEST(*,NTKEEP)
C
C*********************************************************************
        COMMON /FRMTS/ FMT(13),FMTTRI(14)
       COMMON/BIG/B(60,9),SUMXY(60,9),SUMXX(1830),SUMXXI(1830)
      X  ,DUMMY(1)
       DOUBLE PRECISION B,SUMXY,SUMXX,SUMXXI
       COMMON/MED/B0(9),SUMX(69),SUMX2(69),SUMY2(9),ZEAN(69),
      X  CON(99),ERRMS(9),IDENT(13),IDOUT(60),NCON(200),NTERM(69),
      X  NTRAN(100),NXCOD(100),POOLED(9),REPVAR(9),RESMS(9),X(99)
       DOUBLE PRECISION B0,SUMX,SU MX2,SUMY2,ZEAN
       COMMON/SMALL/      BYPASS,BZERO,DELETE, FIRST, IFCHI,  IFSSR,
      X      IFTT,        IFWT,        INPUT,       INPUT5,     INTER,
      X      ISTRAT,      JCOL,        KONNO,       LENGTH,     LIST,
      X      NERROR,      NODEP,       NOOB,                    NOTERM,
      X      NOVAR,       NPDEG,       NRES,        NTRANS,     NWHERE,
      X      P,           PREDCT,      REPS,        RWT,
```

```
     X      STORYI,      STORYC,      STORYX,      TOTWT,       WEIGHT,
     X ERRFXD,  ECONMY,  IOUT,  ICOL
      LOGICAL ECONMY
      LOGICAL SATRTD
      LOGICAL      BYPASS,     BZERO,      DELETE,     IFCHI,
     XIFSSR,      IFTT,       IFWT,       REPS,       PREDCT,
     XSTORYC,     STORYX,     STORYI,     FIRST ,ERRFXD
      DOUBLE PRECISION RWT,TOTWT,WEIGHT
C
C*********************************************************************
      LOGICAL MAKENU,NOZERO
      DIMENSION   T(35,13),PLEVEL(13)
      DIMENSION DEVB(60,9)     , PROB(60,9)      ,TT(60,9)
      EQUIVALENCE (DUMMY(91),DEVB,TT),(DUMMY(650),PROB)
      EQUIVALENCE  (P,PWANT)
C
C*********************************************************************
      DATA  (PLEVEL(JJ),JJ=1,13) /0.10,0.20,0.30,0.40,0.50,0.60,0.70,
     10.80,0.90,0.95,0.98,0.99,0.999 /
      DATA    (T(1,JJ),JJ=1,13) /0.158,0.325,0.510,0.727,1.000,1.376,
     1       1.963,3.078,6.314,12.706,31.821,63.657,636.619  /,
     2         (T(2,JJ),JJ=1,13) /0.142,0.289,0.445,0.617,0.816,1.061,
     3       1.386,1.886,2.920,4.3027,6.965,9.925,31.598      /,
     4         (T(3,JJ),JJ=1,13) /0.137,0.277,0.424,0.584,0.765,0.978,
     5       1.250,1.638,2.353,3.1825,4.541,5.841,12.924      /,
     6         (T(4,JJ),JJ=1,13) /0.134,0.271,0.414,0.569,0.741,0.941,
     7       1.190,1.533,2.132,2.7764,3.747,4.604,8.610       /,
     8         (T(5,JJ),JJ=1,13) /0.132,0.267,0.408,0.559,0.727,0.920,
     9       1.156,1.476,2.015,2.5706,3.365,4.032,6.869       /,
     A         (T(6,JJ),JJ=1,13) /0.131,0.265,0.404,0.553,0.718,0.906,
     B       1.134,1.440,1.943,2.4469,3.143,3.707,5.959       /,
     C         (T(7,JJ),JJ=1,13) /0.130,0.263,0.402,0.549,0.711,0.896,
     D       1.119,1.415,1.895,2.3646,2.998,3.499,5.408       /,
     E         (T(8,JJ),JJ=1,13) /0.130,0.262,0.399,0.546,0.706,0.889,
     F       1.108,1.397,1.860,2.3060,2.896,3.355,5.041       /,
     G         (T(9,JJ),JJ=1,13) /0.129,0.261,0.398,0.543,0.703,0.883,
     H       1.100,1.383,1.833,2.2622,2.821,3.250,4.781       /,
     I        (T(10,JJ),JJ=1,13) /0.129,0.260,0.397,0.542,0.700,0.879,
     J       1.093,1.372,1.812,2.2281,2.764,3.169,4.587       /
      DATA    (T(11,JJ),JJ=1,13) /0.129,0.260,0.396,0.540,0.697,0.876,
     1       1.088,1.363,1.796,2.2010,2.718,3.106,4.437       /,
     2        (T(12,JJ),JJ=1,13) /0.128,0.259,0.395,0.539,0.695,0.873,
     3       1.083,1.356,1.782,2.1788,2.681,3.055,4.318       /,
     4        (T(13,JJ),JJ=1,13) /0.128,0.259,0.394,0.538,0.694,0.870,
     5       1.079,1.350,1.771,2.1604,2.650,3.012,4.221       /,
     6        (T(14,JJ),JJ=1,13) /0.128,0.258,0.393,0.537,0.692,0.868,
     7       1.076,1.345,1.761,2.1448,2.624,2.977,4.140       /,
     8        (T(15,JJ),JJ=1,13) /0.128,0.258,0.393,0.536,0.691,0.866,
     9       1.074,1.341,1.753,2.1315,2.602,2.947,4.073       /,
     A        (T(16,JJ),JJ=1,13) /0.128,0.258,0.392,0.535,0.690,0.865,
     B       1.071,1.377,1.746,2.1199,2.583,2.921,4.015       /,
     C        (T(17,JJ),JJ=1,13) /0.128,0.257,0.392,0.534,0.689,0.863,
     D       1.069,1.333,1.740,2.1098,2.567,2.898,3.965       /,
     E        (T(18,JJ),JJ=1,13) /0.127,0.257,0.392,0.534,0.688,0.862,
     F       1.067,1.330,1.734,2.1009,2.552,2.878,3.922       /,
     G        (T(19,JJ),JJ=1,13) /0.127,0.257,0.391,0.533,0.688,0.861,
     H       1.066,1.328,1.729,2.0930,2.539,2.861,3.883       /,
     I        (T(20,JJ),JJ=1,13) /0.127,0.257,0.391,0.533,0.687,0.860,
     J       1.064,1.325,1.725,2.0860,2.528,2.845,3.850       /
      DATA    (T(21,JJ),JJ=1,13) /0.127,0.257,0.391,0.532,0.686,0.859,
     1       1.063,1.323,1.721,2.0796,2.518,2.831,3.819       /,
     2        (T(22,JJ),JJ=1,13) /0.127,0.256,0.390,0.532,0.686,0.858,
```

```
     3      1.061,1.321,1.717,2.0739,2.508,2.819,3.792        /,
     4       (T(23,JJ),JJ=1,13) /0.127,0.256,0.390,0.532,0.685,0.858,
     5      1.060,1.319,1.714,2.0687,2.500,2.807,3.767        /,
     6       (T(24,JJ),JJ=1,13) /0.127,0.256,0.390,0.531,0.685,0.857,
     7      1.059,1.318,1.711,2.0639,2.492,2.797,3.745        /,
     8       (T(25,JJ),JJ=1,13) /0.127,0.256,0.390,0.531,0.684,0.856,
     9      1.058,1.316,1.708,2.0595,2.485,2.787,3.725        /,
     A       (T(26,JJ),JJ=1,13) /0.127,0.256,0.390,0.531,0.684,0.856,
     B      1.058,1.315,1.706,2.0555,2.479,2.779,3.707        /,
     C       (T(27,JJ),JJ=1,13) /0.127,0.256,0.389,0.531,0.684,0.855,
     D      1.057,1.314,1.703,2.0518,2.473,2.771,3.690        /,
     E       (T(28,JJ),JJ=1,13) /0.127,0.256,0.389,0.530,0.683,0.855,
     F      1.056,1.313,1.701,2.0484,2.467,2.763,3.674        /,
     G       (T(29,JJ),JJ=1,13) /0.127,0.256,0.389,0.530,0.683,0.854,
     H      1.055,1.311,1.699,2.0452,2.462,2.756,3.659        /,
     I       (T(30,JJ),JJ=1,13) /0.127,0.256,0.389,0.530,0.683,0.854,
     J      1.055,1.310,1.697,2.0423,2.457,2.750,3.646        /
      DATA   (T(31,JJ),JJ=1,13) /0.126,0.255,0.388,0.529,0.681,0.851,
     1      1.050,1.303,1.684,2.0211,2.423,2.704,3.551        /,
     2       (T(32,JJ),JJ=1,13) /0.126,0.254,0.387,0.527,0.679,0.848,
     3      1.046,1.296,1.671,2.0003,2.390,2.660,3.460        /,
     4       (T(33,JJ),JJ=1,13) /0.126,0.254,0.386,0.526,0.677,0.845,
     5      1.041,1.289,1.658,1.9799,2.358,2.617,3.373        /,
     6       (T(34,JJ),JJ=1,13) /0.126,0.253,0.385,0.524,0.674,0.842,
     7      1.036,1.282,1.645,1.9600,2.326,2.576,3.291        /
C
C     T(II,JJ)   IS THE T-STATISTIC AT THE TABULATED DEGREES OF FREEDOM
C      (II)  AND AT THE TABULATED PROBABILITY LEVELS (JJ).
C       II=DEGREES OF FREEDOM, EXCEPT FOR
C       II=31 IS FOR 40 DEGREES
C       II=32 IS FOR 60
C       II=33 IS FOR 120
C       II=34 IS FOR INFINITY
C
C     JJ    PROBABILITY LEVEL      *      JJ    PROBABILITY LEVEL
C      1         0.10              *       8         0.80
C      2         0.20              *       9         0.90
C      3         0.30              *      10         0.95
C      4         0.40              *      11         0.98
C      5         0.50              *      12         0.99
C      6         0.60              *      13         0.999
C      7         0.70              *
C
C**********************************************************************
C     CALCULATE T STATISTICS
C
  220 WRITE (LIST,230)
  230 FORMAT(1H0,23HCALCULATED T STATISTICS /75H THE T STATISTICS CAN BE
     1 USED TO TEST THE NET REGRESSION COEFFICIENTS B(I). )
      DO 260 J=1,NOTERM
      DO 240 K=1,NODEP
      TT(J,K)=ABS(B(J,K)/DEVB(J,K))
  240 CONTINUE
      WRITE (LIST,250)(TT(J,K),K=1,NODEP)
  250 FORMAT(1H 9G14.6)
  260 CONTINUE
C
C**********************************************************************
      NDEG = NERROR
C
C**********************************************************************
C     SEARCH THE TABLE OF TABULATED DEGREES OF FREEDOM
```

```
C
      MAKENJ=.FALSE.
      IF(NDEG-30)290,290,300
  290 II=NDEG
      GO TO 400
  300 IF(NDEG-40)310,320,330
  310 FINV=1.0/40.0
      FM1INV=1.0/30.0
      MAKENJ=.TRUE.
  320 II=31
      GO TO 400
  330 IF(NDEG-60)340,350,360
  340 FINV=1.0/60.0
      FM1INV=1.0/40.0
      MAKENJ=.TRUE.
  350 II=32
      GO TO 400
  360 IF(NDEG-120)370,380,390
  370 FINV=1.0/120.0
      FM1INV=1.0/60.0
      MAKENJ=.TRUE.
  380 II=33
      GO TO 400
  390 II=34
      FINV=0.0
      FM1INV=1.0/120.0
      MAKENU=.TRUE.
C
C
  400 WRITE(LIST,410)
  410 FORMAT(104H UNDER NULL HYPOTHESIS THE INTERVAL (-T,T) WHERE T IS G
     XIVEN ABOVE, HAS APPROX PROBABILITY LISTED BELOW.      /42H  MINUS S
     XIGN INDICATES PROB EXCEEDS .999.    )
      IF(.NOT.MAKENU) GO TO 430
      FNDEG=NDEG
      DO  420   JJ=1,13
      T(35,JJ)=T(II,JJ)+((1.0/FNDEG - FINV)/(FM1INV-FINV))*(T(II-1,JJ)
     1 -T(II,JJ))
  420 CONTINUE
      II=35
  430 DO  550   J=1,NOTERM
      DO  540   K=1,NODEP
      DO 440   JJ=1,13
      JJ=JJ
      IF(T(II,JJ)-TT(J,K))440,450,460
  440 CONTINUE
      PROB(J,K)=-0.999
      GO TO 540
  450 PROB(J,K)=PLEVEL(JJ)
      GO TO 540
  460 IF(JJ.LE.9) GO TO 470
      JJ1=JJ-2
      JJ2=JJ-1
      JJ3=JJ
      GO TO 490
  470 IF(JJ.LE.4)GO TO 480
      JJ1=JJ-1
      JJ2=JJ
      JJ3=JJ+1
      GO TO 490
  480 JJ1=JJ
      JJ2=JJ+1
      JJ3=JJ+2
```

```
C
C     PERFORM A THREE-POINT LAGRANGE INTERPOLATION
C
  490 X=ALOG(TT(J,K))
      X1=ALOG(T(II,JJ1))
      X2=ALOG(T(II,JJ2))
      X3=ALOG(T(II,JJ3))
      IF(TT(J,K).LE.1.0) GO TO 500
      Y1=ALOG(1.0-PLEVEL(JJ1))
      Y2=ALOG(1.0-PLEVEL(JJ2))
      Y3=ALOG(1.0-PLEVEL(JJ3))
      GO TO 510
  500 Y1=ALOG(PLEVEL(JJ1))
      Y2=ALOG(PLEVEL(JJ2))
      Y3=ALOG(PLEVEL(JJ3))
  510 PROB(J,K)= ((X-X2)*(X-X3)*Y1)/((X1-X2)*(X1-X3)) + ((X-X1)*(X-X3)
     1  *Y2)/((X2-X1)*(X2-X3))  + ((X-X1)*(X-X2)*Y3)/((X3-X1)*(X3-X2))
      IF(TT(J,K)-1.0) 520,520,530
  520 PROB(J,K)=EXP(PROB(J,K))
      GO TO 540
  530 PROB(J,K)=1.0-EXP(PROB(J,K))
  540 CONTINUE
C*********************************************************************
C     WRITE THE PROBABILITIES (1.0-ALPHA)
      WRITE(LIST,550) IDOUT(J),(PROB(J,K),K=1,NODEP)
  550 FORMAT(1H I3,9(8X,F6.3))
  560 CONTINUE
C
C*********************************************************************
C     LIST THE DESIRED VALUE OF PROBABILITY (PWANT)
C
  570 PERCEN=PWANT*100.0
      WRITE(LIST,580) PERCEN
  580 FORMAT(1H0,36HTHE DESIRED VALUE OF PROBABILITY IS ,F5.1, 8H PERCEN
     1T )
C
C     DELETE THE TERM WITH THE LOWEST COMPUTED PROBABILITY IF THAT
C     PROBABILITY IS LESS THAN THAT DESIRED (PWANT)
C
      IF(.NOT.DELETE) GO TO 660
      IF(NTKEEP.EQ.NOTERM) GO TO 660
      IOUT=0
  590 AMIN=PWANT
      JLO=MAX0(1,NTKEEP)
      DO 620 J=JLO,NOTERM
      IF(ABS(PROB(J,1))-PWANT)600,620,620
  600 IF(ABS(PROB(J,1))-AMIN)610,620,620
  610 AMIN=ABS(PROB(J,1))
      IOUT=J
  620 CONTINUE
      IF(IOUT) 660,660,630
  630 WRITE(LIST,650) IDOUT(IOUT)
  650 FORMAT(1H 10X,11HTHE TERM X(,I2,18H) IS BEING DELETED   )
      GO TO 670
C      ALL VARIABLES REMAINING HAVE BEEN CONCLUDED SIGNIFICANT
  660 RETURN1
  670 RETURN
      END
```

```
$IBFTC RSTATX

C
C      SUBROUTINE RSTATS
C
C      PURPOSE
C          1)COMPUTE AND PRINT THE ANALYSIS OF VARIANCE TABLES ON
C            REGRESSION AND LACK-OF-FIT IF APPROPRIATE.
C          2)COMPUTE AND PRINT R-SQUARED AND STANDARD ERROR OF
C            ESTIMATE
C          3)COMPUTE AND PRINT SUMS OF SQUARES DUE TO EACH VARIABLE
C            IF IT WERE LAST TO ENTER REGRESSION
C          4)COMPUTE AND PRINT THE STANDARD DEVIATIONS OF EACH
C            REGRESSION COEFFICIENT.
C
      SUBROUTINE RSTATS(IORDER)
C****************************************
      COMMON/BIG/B(60,9),SUMXY(60,9),SUMXX(1830),SUMXXI(1830)
     X  ,DUMMY(1)
      DOUBLE PRECISION B,SUMXY,SUMXX,SUMXXI
      COMMON/MED/BO(9),SUMX(69),SUMX2(69),SUMY2(9),ZEAN(69),
     X  CON(99),ERRMS(9),IDENT(13),IDOUT(60),NCON(200),NTERM(69),
     X  NTRAN(100),NXCOD(100),POOLED(9),REPVAR(9),RESMS(9),X(99)
      DOUBLE PRECISION BO,SUMX,SU MX2,SUMY2,ZEAN
       COMMON /FRMTS/ FMT(13),FMTTRI(14)
      COMMON/SMALL/     BYPASS,BZERO,DELETE, FIRST, IFCHI,  IFSSR,
     X       IFTT,       IFWT,       INPUT,      INPUT5,     INTER,
     X       ISTRAT,     JCOL,       KONNO,      LENGTH,     LIST,
     X       NERROR,     NODEP,      NOOB,                   NOTERM,
     X       NOVAR,      NPDEG,      NRES,       NTRANS,     NWHERE,
     X       P,          PREDCT,     REPS,       RWT,
     X       STORYI,      STORYC,     STORYX,     TOTWT,      WEIGHT,
     X ERRFXD,  ECONMY,  IOUT,  ICOL
      LOGICAL ECONMY
      LOGICAL SATRTD
      LOGICAL      BYPASS,     BZERO,      DELETE,     IFCHI,
     XIFSSR,       IFTT,       IFWT,       REPS,       PREDCT,
     XSTORYC,      STORYX,     STORYI,     FIRST ,ERRFXD
      DOUBLE PRECISION RWT,TOTWT,WEIGHT
C
C*******************************************************************
      DIMENSION SSQREG(9),    SSQRES(9),     REGMS(9),
     X       XLOF(9),     XLOFMS(9),    FRATIO(9),  RSQD(9),     R(9),
     X       SSQLST(9),   DEVB(60,9)
      EQUIVALENCE (DUMMY(10),SSQRES),(DUMMY(19),REGMS),(DUMMY(37),XLOF)
     X ,(DUMMY(46),XLOFMS),(DUMMY(55),FRATIO),(DUMMY(64),RSQD),
     X (DUMMY(73),R),(DUMMY(82),SSQLST),(DUMMY(91),DEVB),
     X  (DUMMY(700),SSQREG)
      DOUBLE PRECISION SSQREG
C
C*******************************************************************
C     COMPUTE DEGREES OF FREEDOM AND RECIPROCALS
      NREG= IORDER
      NTOT= IFIX(TOTWT)-1
      IF(.NOT.BZERO) NTOT= NTOT+1
      NRES= NTOT-NREG
      NLOF= NRES - NPDEG
      RNREG= 1.0/FLOAT(NREG)
      IF(NRES.EQ.0) GO TO 980
      RNRES= 1.0/FLOAT(NRES)
      SATRTD=.FALSE.
      IF(NLOF.EQ.0) GO TO 90
```

```
      RNLOF=1.0/FLOAT(NLOF)
      GO TO 100
   90 SATRTD=.TRUE.
  100 CONTINUE
      NXTERM=IORDER
      RNOOB=RWT
C
C**********************************************************************
C      COMPUTE RESIDUAL SUM OF SQUARES, RESIDUAL VARIANCE, VARIANCE
C      FROM REPLICATIONS IF APPROPRIATE, AND THE F-RATIO OF MEAN SQUARE
C     LACK-OF-FIT AND MEAN SQUARE RESIDUALS.
      DO 210 J=1,NODEP
      SSQREG(J)=0.0
      DO 200 I=1,NXTERM
      SSQREG(J)=SSQREG(J) + B(I,J)*SUMXY(I,J)
  200 CONTINUE
      SSQRES(J)=SUMY2(J)-SSQREG(J)
      REGMS(J)= SSQREG(J)* RNREG
      RESMS(J)= SSQRES(J)*RNRES
      RSQD(J)=SSQREG(J)/SUMY2(J)
      R(J)=SQRT(RSQD(J))
      IF((.NOT.REPS).OR.SATRTD) GO TO 210
      XLOF(J)=SSQRES(J)-POOLED(J)
      XLOFMS(J)= XLOF(J)*RNLOF
      FRATIO(J)=XLOFMS(J)/REPVAR(J)
  210 CONTINUE
C
C**********************************************************************
C      DETERMINE WHICH ESTIMATE OF SIGMA SQUARED SHOULD BE USED IN
C      HYPOTHESIS TESTS.  PUT THE PROPER ONE IN ERRMS AND SET ERRFXD
C      TO TRUE IF THE PRESENT VALUE IS TO BE USED FOR ALL FOLLOWING
C      TESTS AND T-STATISTICS.
      IOUT=ISTRAT
      IF(ERRFXD) GO TO 250
      IF(ISTRAT.NE.3) GO TO 214
  211 DO 213 J=1,NODEP
  213 ERRMS(J)= RESMS(J)
      NERROR = NRES
      IOUT=3
      GO TO 250
  214 IF(ISTRAT.NE.1) GO TO 218
      IF(.NOT.REPS) GO TO 211
      DO 215 J=1,NODEP
  215 ERRMS(J)= REPVAR(J)
      NERROR= NPDEG
      ERRFXD= .TRUE.
      IOUT=1
      GO TO 250
  218 IF(FIRST.AND.(IORDER.EQ.NOTERM)) GO TO 220
      GO TO 211
  220 ERRFXD= .TRUE.
      DO 222 J=1,NODEP
  222 ERRMS(J)= RESMS(J)
      NERROR= NRES
      ISTRAT=2
      IOUT=2
C
C**********************************************************************
C     WRITE ANOVA TABLES
  250 DO 500 J=1,NODEP
      IF(ERRMS(J).EQ.0.0) ERRMS(J)=1.0E-30
      WRITE(LIST,1001) J
```

```
      WRITE(LIST,1002)
      WRITE(LIST,1003) SSQREG(J), NREG, REGMS(J)
      WRITE(LIST,1004) SSQRES(J), NRES, RESMS(J)
      WRITE(LIST,1005)
      WRITE(LIST,1006) SUMY2(J),NTOT
      WRITE(LIST,1007)
      WRITE(LIST,1500) RSQD(J), R(J)
      STD=SQRT(RESMS(J))
      WRITE(LIST,1600) STD
      WRITE(LIST,1700) IOUT,ERRMS(J),NERROR
      F=REGMS(J)/ERRMS(J)
      WRITE(LIST,1750)F,NREG,NERROR
      IF((.NOT.REPS).OR.SATRTD) GO TO 500
      WRITE(LIST,2001)
      WRITE(LIST,1002)
      WRITE(LIST,2005) XLOF(J), NLOF, XLOFMS(J)
      WRITE(LIST,2006) POOLED(J), NPDEG, REPVAR(J)
       WRITE(LIST,1004) SSQRES(J),NRES,RESMS(J)
      WRITE(LIST,1005)
      WRITE(LIST,2008) FRATIO(J)
      WRITE(LIST,1007)
  500 CONTINUE
C
C*********************************************************************
C     COMPUTE CONTRIBUTION OF EACH INDEPENDENT VARIABLE TO REG SUM
C     OF SQUARES AS IF IT WERE LAST TO ENTER
      WRITE(LIST,370)
      IR= 0
      DO 8635 K=1,NXTERM
      IR= IR+K
      DO 8632 J=1,NODEP
 8632 SSQLST(J)= B(K,J)**2/SUMXXI(IR)
      WRITE(LIST,380) IDOUT(K),(SSQLST(J),J=1,NODEP)
 8635 CONTINUE
C
C*********************************************************************
C     COMPUTE STANDARD DEVIATION OF REGRESSION COEFFICIENTS
      WRITE(LIST,375)
      IF(.NOT.BZERO) GO TO 959
      DO 910 J=1,NXTERM
      R(J)=0.0
      DO 910 I=1,NXTERM
      CALL LOC(I,J,IR)
      R(J)=R(J)+ZEAN(I)*SUMXXI(IR)
  910 CONTINUE
      XXT=0.0
      DO 920 J=1,NXTERM
  920 XXT=XXT+ZEAN(J)*R(J)
      DO 930 K=1,NODEP
  930 DEVB(1,K)=SQRT(ERRMS(K)*(RNOOB+XXT))
      K=0
      WRITE(LIST,380) K,(DEVB(1,J),J=1,NODEP)
  959 IR=0
      DO 970 J=1,NXTERM
      IR= IR+J
      DO 960 K=1,NODEP
      DEVB(J,K) =SQRT(ERRMS(K)*SUMXXI(IR))
  960 CONTINUE
       WRITE(LIST,380) IDOUT(J),(DEVB(J,KR),KR=1,NODEP)
  970 CONTINUE
      RETURN
C
```

```
C***********************************************************************
C      FORMATS
 1001 FORMAT(42H4ANOVA OF REGRESSION ON DEPENDENT VARIABLE  I5)
 1002 FORMAT(1H 79(1H*)/79H        SOURCE           SUMS OF SQUARES     DEG
     XREES OF FREEDOM  MEAN SQUARES           /1H 79(1H-))
 1003 FORMAT(17H      REGRESSION       G20.8, 5X,I10,5X,G20.8)
 1004 FORMAT(17H      RESIDUAL         G20.8, 5X,I10,5X,G20.8)
 1005 FORMAT(1H 79(1H-))
 1006 FORMAT(17H      TOTAL            G20.8, 5X,I10)
 1007 FORMAT(1H 79(1H*))
 2001 FORMAT(1X/1X/22H  ANOVA OF LACK OF FIT   )
 2005 FORMAT(17H      LACK OF FIT      G20.8, 5X,I10,5X,G20.8)
 2006 FORMAT(17H      REPLICATION      G20.8, 5X,I10,5X,G20.8)
 2008 FORMAT(28H      F = MS(LOF)/MS(REPS) =  F10.3 )
  370 FORMAT(74H1 SUMS OF SQUARES DUE TO EACH VARIABLE IF IT WERE LAST T
     XO ENTER REGRESSION )
  375 FORMAT(115H2 STANDARD DEVIATION OF REGRESSION COEFFICIENTS (DERIVE
     XD FROM DIAGONAL ELEMENTS OF  (X TRANSPOSE X)INVERSE MATRIX ))
  380 FORMAT(1H I3,9G14.6)
 1500 FORMAT(40H        R SQUARED = SSQ(REG) / SSQ(TOT) =    F8.6,
     X 5X, 4HR = F7.6)
 1600 FORMAT(34H       STANDARD ERROR OF ESTIMATE   G14.6)
 1700 FORMAT(24H USING POOLING STRATEGY I2,25H THE ERROR MEAN SQUARE =
     X G14.7, 26H WITH DEGREES OF FREEDOM = I6)
 1750 FORMAT(5X,19HF=MS(REG)/MS(ERR)= F6.2,5X,13HCOMPARE TO F(I2,1H,I3,1
     XH))
  980 WRITE(LIST,981)
  981 FORMAT(41H ZERO RESIDUAL DEGREES OF FREEDOM. STOP.    )
      STOP
      END
```

```
$IBFTC RECTX

      SUBROUTINE RECT(IROW,JJCOL,IMAX,JMAX,A,B,FMT,II)
      DIMENSION  A(IMAX,JMAX),FMT(14),XOUT(8)
      DOUBLE PRECISION B,DXOUT
      DIMENSION B(IMAX,JMAX),DXOUT(8)
      COMMON/SMALL/DUM(15),LIST
      DATA J8/8/
      LOGICAL OUT
      OUT =.FALSE.
      JTIMES=0
      JCOL=JJCOL
    5 JNXT=JCOL-J8
      IF(JNXT) 10,20,30
   10 JP=JCOL
      GO TO 40
   20 JP=J8
      GO TO 40
   30 JCOL=JNXT
      JP=J8
      GO TO 50
   40 OUT=.TRUE.
   50 DO 100 I=1,IROW
      GO TO (55,75),II
   55 CONTINUE
      DO 60 J=1,JP
      JJ=JTIMES +J
```

```
   60 XOUT(J) = A(I,JJ)
      WRITE(LIST,FMT) I,(XOUT(K),K=1,JP)
      GO TO 100
   75 DO 80 J=1,JP
      JJ=JTIMES+J
   80 DXOUT(J)=B(I,JJ)
      WRITE(LIST,FMT) I,(DXOUT(J),J=1,JP)
  100 CONTINUE
      IF(OUT) RETURN
      WRITE(LIST,110)
  110 FORMAT(1H /1H )
      JTIMES=JTIMES +JP
      GO TO 5
      END
```

```
$IBFTC PREDIX

C
C      SUBROUTINE PREDIC
C
C      PURPOSE
C           1)READ INPUT LEVELS OF INDEPENDENT VARIABLES AND COMPUTE
C             A PREDICTED RESPONSE FROM THE ESTIMATED REGRESSION EQUATION.
C           2)COMPUTE VARIANCE AND STADARD DEVIATION OF THE PREDICTED
C             MEAN VALUE AND A SINGLE FURTHER OBSERVATION.
C
C      SUBROUTINES NEEDED
C             TRANS
C             LOC
C
C      REMARKS
C             VALUES FOR DEPENDENT VARIABLES ARE NOT NECCESARY FOR THE
C             PREDICTING OF VALUES. HOWEVER, A DUMMY VALUE MAY NEED TO
C             BE SUPPLIED IF A ZERO (BLANK) INPUT VALUE WILL CAUSE AN
C             IMPOSSIBLE OPERATION TO BE ATTEMPTED DURING THE
C             TRANSFORMATIONS.
C*********************************************************************
C
      SUBROUTINE PREDIC
       COMMON /FRMTS/ FMT(13),FMTTRI(14)
      COMMON/BIG/B(60,9),SUMXY(60,9),SUMXX(1830),SUMXXI(1830)
      DOUBLE PRECISION B,SUMXY,SUMXX,SUMXXI
      COMMON/MED/B0(9),SUMX(69),SUMX2(69),SUMY2(9),ZEAN(69),
     X  CON(99),ERRMS(9),IDENT(13),IDOUT(60),NCON(200),NTERM(69),
     X  NTRAN(100),NXCOD(100),POOLED(9),REPVAR(9),RESMS(9),X(99)
      DOUBLE PRECISION B0,SUMX,SU MX2,SUMY2,ZEAN
      COMMON/SMALL/     BYPASS,BZERO,DELETE, FIRST, IFCHI,  IFSSR,
     X       IFTT,        IFWT,         INPUT,        INPUT5,      INTER,
     X       ISTRAT,      JCOL,         KONNO,        LENGTH,      LIST,
     X       NERROR,      NODEP,        NOOB,                      NOTERM,
     X       NOVAR,       NPDEG,        NRES,         NTRANS,      NWHERE,
     X       P,           PREDCT,       REPS,         RWT,
     X       STORYI,       STORYC,       STORYX,       TOTWT,        WEIGHT,
     X ERRFXD,  ECONMY,  IOUT,  ICOL
      LOGICAL ECONMY
      LOGICAL      BYPASS,      BZERO,       DELETE,      IFCHI,
     XIFSSR,       IFTT,        IFWT,        REPS,        PREDCT,
     XSTORYC,      STORYX,      STORYI,      FIRST ,ERRFXD
```

```
      DOUBLE PRECISION WEIGHT,RWT,TOTWT
      DIMENSION YCALC(9),      V(60),       VARM(9),       SEEM(9),
     X     VARP(9),      SEEP(9)
      DOUBLE PRECISION XXT,V
      EQUIVALENCE  (YCALC(1),SUMXX(1)),   (V(1),SUMXX(150)),
     X   (VARM(1),SUMXX(71)),    (SEEM(1),SUMXX(80)), (VARP(1),SUMXX(89))
     X   ,(SEEP(1),SUMXX(98))
      EQUIVALENCE (RNOOB,RWT)
C
C***********************************************************************
C
      IF(NOTERM.EQ.0) RETURN
      WRITE(LIST,3)
      READ(INPUT5,5) NPRED
C
      DO 500 KK=1,NPRED
C
  105 READ(INPUT5,FMT)(X(I),I=1,ICOL)
      WRITE(LIST,110)(X(I),I=1,ICOL)
  125 CALL TRANS
      DO 130 K=1,JCOL
      I=NTERM(K)
      X(K) = CON(I)
  130 CONTINUE
      WRITE(LIST,135)(X(I),I=1,NOTERM)
C
C     COMPUTE PREDICTED RESPONSE
  140 DO 150 K=1,NODEP
      YCALC(K) = BO(K)
      IF(.NOT.BZERO) YCALC(K)=0.0
      DO 150 J=1,NOTERM
      YCALC(K)= YCALC(K) + B(J,K)*X(J)
  150 CONTINUE
C
C     COMPUTE VARIANCE AND STANDARD DEVIATION OF REGRESSION LINE
C     AND VARIANCE AND STANDARD DEVIATION OF PREDICTED VALUE
C     AT THE POINT XO
C
      DO 250 K=1,NOTERM
      V(K)=0.0D0
      DO 250 J=1,NOTERM
      CALL LOC(J,K,IR)
      V(K)=V(K) + (X(J)-ZEAN(J))*SUMXXI(IR)
  250 CONTINUE
      XXT=0.0D0
      DO 275 K=1,NOTERM
      XXT = XXT + (X(K)-ZEAN(K))*V(K)
  275 CONTINUE
      XRNOOB = RNOOB
      IF(.NOT.BZERO) XRNOOB=0.0
      DO 300 K=1,NODEP
      VARM(K)= ERRMS(K)*(XRNOOB + XXT)
      SEEM(K)=SQRT(VARM(K))
      VARP(K)= ERRMS(K)+VARM(K)
      SEEP(K)=SQRT(VARP(K))
  300 CONTINUE
      WRITE(LIST,310)(YCALC(K),K=1,NODEP)
      WRITE(LIST,320)(VARM (K),K=1,NODEP)
      WRITE(LIST,320)(SEEM (K),K=1,NODEP)
      WRITE(LIST,320)(VARP (K),K=1,NODEP)
      WRITE(LIST,320)(SEEP (K),K=1,NODEP)
C
```

```
C
  500 CONTINUE
      RETURN
    3 FORMAT(54H1FOR EACH SET OF INDEP VARIABLES THERE IS COMPUTED...  /
     X  20H PREDICTED RESPONSE                                         /
     X 29H VARIANCE OF REGRESSION LINE                                 /
     X  34H STANDARD DEVIATION OF REGRESSION                           /
     X  29H VARIANCE OF PREDICTED VALUE                                /
     X  39H STANDARD DEVIATION OF PREDICTED VALUE      )
    5 FORMAT(I4)
  110 FORMAT(39HKINPUT DATA FOR THIS PREDICTED RESPONSE /(1H 9G14.6))
  135 FORMAT(56HK INDEPENDENT TERMS ACCORDING TO FINAL REGRESSION MODEL
     X  /(1H 9G14.6))
  310 FORMAT( 55HKPREDICTED RESPONSE FOR ABOVE INDEP VARIABLES
     X    /(1H 9G14.6))
  320 FORMAT(1H 9G14.6)
      END
```

```
$IBFTC SUMUPX
C
C      SUBROUTINE SUMUPS
C
C      PURPOSE
C           1)CALCULATE (X TRANSPOSE X) AND (X TRANSPOSE Y) MATRICES ONE
C             OBSERVATION AT A TIME.
C           2)COMPUTE TOTAL OF THE WEIGHTS
C           ** BOTH CALCULATIONS ARE IN DOUBLE PRECISION
C
C      SUBROUTINES NEEDED
C             LOC
C
C*************************************************************************
       SUBROUTINE SUMUP
       COMMON/BIG/B(60,9),SUMXY(60,9),SUMXX(1830),SUMXXI(1830)
       DOUBLE PRECISION B,SUMXY,SUMXX,SUMXXI
       COMMON/MED/BO(9),SUMX(69),SUMX2(69),SUMY2(9),ZEAN(69),
      X  CON(99),ERRMS(9),IDENT(13),IDOUT(60),NCON(200),NTERM(69),
      X  NTRAN(100),NXCOD(100),POOLED(9),REPVAR(9),RESMS(9),X(99)
       DOUBLE PRECISION BO,SUMX,SU MX2,SUMY2,ZEAN
       COMMON/SMALL/     BYPASS,BZERO,DELETE, FIRST, IFCHI,  IFSSR,
      X       IFTT,       IFWT,        INPUT,       INPUT5,     INTER,
      X       ISTRAT,     JCOL,        KONNO,       LENGTH,     LIST,
      X       NERROR,     NODEP,       NOOB,                    NOTERM,
      X       NOVAR,      NPDEG,       NRES,        NTRANS,     NWHERE,
      X       P,          PREDCT,      REPS,        RWT,
      X       STORYI,      STORYC,      STORYX,      TOTWT,       WEIGHT,
      X ERRFXD,  ECONMY,  IOUT,  ICOL
       LOGICAL ECONMY
       LOGICAL      BYPASS,      BZERO,       DELETE,      IFCHI,
      XIFSSR,      IFTT,        IFWT,        REPS,        PREDCT,
      XSTORYC,     STORYX,      STORYI,      FIRST ,ERRFXD
       DOUBLE PRECISION RWT,TOTWT,WEIGHT
        DOUBLE PRECISION DUB1,DUB2
C
C*********************************************************************
       DO 110 I=1,JCOL
       SUMX(I)=SUMX(I)+X(I)*WEIGHT
```

```
  110 CONTINUE
      IR=0
      DO 100 K=1,NOTERM
      DUB1=X(K)
      DO 90 J=1,NODEP
      KBAR=J+NOTERM
      DUB2=X(KBAR)
      SUMXY(K,J)=SUMXY(K,J)+DUB1*DUB2*WEIGHT
   90 CONTINUE
      DO 50 I=1,K
C
      IR=IR+1
      DUB2=X(I)
      SUMXX(IR)=SUMXX(IR)+DUB1*DUB2*WEIGHT
   50 CONTINUE
  100 CONTINUE
      DO 15 J=1,NODEP
      KBAR=NOTERM + J
      DUB1=X(KBAR)
      SUMY2(J)=SUMY2(J)+DUB1*DUB1*WEIGHT
   15 CONTINUE
      TOTWT=TOTWT+WEIGHT
      RETURN
      END
```

```
$IBFTC BORDXX

C
C     SUBROUTINE BORD
C
C     PURPOSE
C           TO COMPLETE THE INVERSION OF A SYMMETRIC POSITIVE DEFINITE
C           MATRIX A OF ORDER N GIVEN THAT THE UPPER LEFT SUB-
C           MATRIX OF ORDER N-1 HAS ALREADY BEEN INVERTED.
C
C     SUBROUTINES NEEDED
C           LOC
C
C     REMARKS
C           ONLY THE UPPER TRIANGULAR PART OF A IS STORED AS A
C           VECTOR IN THE ORDER A(1,1),A(1,2),A(2,2),A(1,3),...ETC
      SUBROUTINE BORD(IORDER,A)
C
C
      DIMENSION  BETA(60),A(1)
      DOUBLE PRECISION A,ALPHA,RALPHA ,BETA
C
C
      ALPHA= 0.0D0
      NM1= IORDER-1
      IF(NM1) 100,100,200
  100 A(1) = 1.0/A(1)
      GO TO 600
  200 M=NM1*(NM1+1)/2
      LEN = M + IORDER
C
      DO 400 I=1,NM1
```

```
      BETA(I)= 0.0D0
      MI= M+I
      DO 350 J=1,NM1
      CALL LOC(I,J,II)
      MJ= M+J
      BETA(I)= BETA(I)-A(II)*A(MJ)
  350 CONTINUE
      ALPHA= ALPHA + A(MI)*BETA(I)
  400 CONTINUE
C
C
       ALPHA = ALPHA + A(LEN)
      RALPHA=1.0D0/ALPHA
      A(LEN) = RALPHA
C
      DO 500 I=1,NM1
      DO 500 J=1,I
      CALL LOC(I,J,II)
      A(II)= A(II) + BETA(I)*BETA(J)*RALPHA
  500 CONTINUE
C
      DO 550 J=1,NM1
      MJ= M+J
      A(MJ)= BETA(J)*RALPHA
  550 CONTINUE
C
C
  600 CONTINUE
      RETURN
      END
```

```
$IBFTC MFIXXX

      SUBROUTINE MFIX
C
C*********************************************************************
       COMMON /FRMTS/ FMT(13),FMTTRI(14)
      COMMON/BIG/B(60,9),SUMXY(60,9),SUMXX(1830),SUMXXI(1830)
      DOUBLE PRECISION B,SUMXY,SUMXX,SUMXXI
      COMMON/MED/BO(9),SUMX(69),SUMX2(69),SUMY2(9),ZEAN(69),
     X  CON(99),ERRMS(9),IDENT(13),IDOUT(60),NCON(200),NTERM(69),
     X  NTRAN(100),NXCOD(100),POOLED(9),REPVAR(9),RESMS(9),X(99)
      DOUBLE PRECISION BO,SUMX,SU MX2,SUMY2,ZEAN
      COMMON/SMALL/      BYPASS,BZERO,DELETE, FIRST, IFCHI,  IFSSR,
     X       IFTT,       IFWT,        INPUT,       INPUT5,     INTER,
     X       ISTRAT,     JCOL,        KONNO,       LENGTH,     LIST,
     X       NERROR,     NODEP,       NOOB,                    NOTERM,
     X       NOVAR,      NPDEG,       NRES,        NTRANS,     NWHERE,
     X       P,          PREDCT,      REPS,        RWT,
     X       STORYI,      STORYC,      STORYX,      TOTWT,       WEIGHT,
     X ERRFXD,  ECONMY,  IOUT,  ICOL
      LOGICAL ECONMY
      DOUBLE PRECISION RWT,TOTWT,WEIGHT
      LOGICAL      BYPASS,      BZERO,       DELETE,      IFCHI,
     XIFSSR,      IFTT,        IFWT,        REPS,        PREDCT,
     XSTORYC,     STORYX,      STORYI,      FIRST ,ERRFXD
C*********************************************************************
```

```
C
      IF(ECONMY) GO TO 500
      IF(BZERO) GO TO 500
      WRITE(LIST,530)
      WRITE (LIST,540) (SUMX (I),I=1,JCOL)
      WRITE(LIST,560)
      CALL TRIANG(X,SUMXX,NOTERM,8,FMTTRI,2)
      WRITE(LIST,565)
      CALL RECT(NOTERM,NODEP,60,9,X,SUMXY,FMTTRI,2)
C*********************************************************************
C     COMPUTE AND PRINT MEANS. COMPUTE AND PRINT THE(X TRANSPOSE X)
C     MATRIX IN TERMS OF DEVIATIONS FROM MEAN. THE DEVIATIONS FORM
C     OF (X T X)  IS THE VARIANCE-COVARIANCE MATRIX OF THE
C     INDEPENDENT VARIABLES.
  500 CONTINUE
      RWT=1.0D0/TOTWT
      DO 570 I=1,JCOL
  570 ZEAN(I)=SUMX(I)*RWT
      WRITE(LIST,580)
      WRITE(LIST,540) (ZEAN(I),I=1,JCOL)
      IR = 0
      DO 600 J=1,NOTERM
      IR=IR + J
      IF(.NOT.BZERO) GO TO 601
      SUMX2(J)=SUMXX(IR)-SUMX(J)**2 *RWT
      GO TO 600
  601 SUMX2(J)=SUMXX(IR)
  600 CONTINUE
  602 CONTINUE
      IR=1
      DO 620 J=1,NOTERM
      DO 618 K=1,NODEP
      IF(.NOT.BZERO) GO TO 618
      KBAR=NOTERM+K
      SUMXY(J,K)=SUMXY(J,K)-SUMX(J)*SUMX(KBAR)*RWT
  618 CONTINUE
  619 DO 620 K=1,J
      IF(.NOT.BZERO) GO TO 6191
      SUMXX(IR)=SUMXX(IR)-SUMX(K)*SUMX(J)*RWT
 6191 SUMXXI(IR)=SUMXX(IR)/DSQRT(SUMX2(J)*SUMX2(K))
 6194 IR=IR+1
  620 CONTINUE
      IF(.NOT.BZERO) GO TO 6220
      DO 6210 J=1,NODEP
      K=NOTERM+J
       SUMY2(J)=SUMY2(J)-SUMX(K)**2*RWT
 6210 CONTINUE
 6220 CONTINUE
C*********************************************************************
      IF(ECONMY) GO TO 622
      IF(.NOT.BZERO) GO TO 621
      WRITE(LIST,625)
      CALL TRIANG(X,SUMXX,NOTERM,8,FMTTRI,2)
      WRITE(LIST,630)
      CALL RECT(NOTERM,NODEP,60,9,X,SUMXY,FMTTRI,2)
  621 WRITE(LIST,670)
      CALL TRIANG(X,SUMXXI,NOTERM,8,FMTTRI,2)
 622  CONTINUE
C*********************************************************************
      RETURN
C
  530 FORMAT(1H0,32H SUMS OF INDEP AND DEP VARIABLES )
```

```
  540 FORMAT(1H  8G15.7)
  560 FORMAT(21H2X TRANSPOSE X MATRIX     )
  565 FORMAT(21H2X TRANSPOSE Y MATRIX     )
  580 FORMAT(33H MEANS OF INDEP AND DEP VARIABLES     )
  625 FORMAT(53H2X TRANSPOSE X MATRIX WHERE X IS DEVIATION FROM MEAN  )
  630 FORMAT(60H2X TRANSPOSE Y MATRIX WHERE X AND Y ARE DEVIATIONS FROM
     XMEAN  )
  670 FORMAT(25H2CORRELATION COEFFICIENTS    )
      END
```

```
$IBFTC LOCXXX

      SUBROUTINE LOC(I,J,IR)
      IX= I
      JX= J
   20 IF(IX-JX) 22,24,24
   22 IRX= IX + (JX*JX-JX)/2
      GO TO 36
   24 IRX= JX + (IX*IX - IX)/2
   36 IR= IRX
      RETURN
      END
```

```
$IBFTC XTRANS

      SUBROUTINE TRANS
C***********************************************************************
C
      COMMON/BIG/B(60,9),SUMXY(60,9),SUMXX(1830),SUMXXI(1830)
      DOUBLE PRECISION B,SUMXY,SUMXX,SUMXXI
      COMMON/MED/BO(9),SUMX(69),SUMX2(69),SUMY2(9),ZEAN(69),
     X  CON(99),ERRMS(9),IDENT(13),IDOUT(60),NCON(200),NTERM(69),
     X  NTRAN(100),NXCOD(100),POOLED(9),REPVAR(9),RESMS(9),X(99)
      DOUBLE PRECISION BO,SUMX,SUMX2,SUMY2,ZEAN
      COMMON/SMALL/     BYPASS,BZERO,DELETE, FIRST, IFCHI,  IFSSR,
     X       IFTT,         IFWT,         INPUT,        INPUT5,      INTER,
     X       ISTRAT,       JCOL,         KONNO,        LENGTH,      LIST,
     X       NERROR,       NODEP,        NOOB,                      NOTERM,
     X       NOVAR,        NPDEG,        NRES,         NTRANS,      NWHERE,
     X       P,            PREDCT,       REPS,         RWT,
     X       STORYI,        STORYC,       STORYX,       TOTWT,       WEIGHT,
     X ERRFXD,  ECONMY,  IOUT,  ICOL
      LOGICAL ECONMY
      LOGICAL      BYPASS,      BZERO,       DELETE,      IFCHI,
     XIFSSR,       IFTT,        IFWT,        REPS,        PREDCT,
     XSTORYC,      STORYX,      STORYI,      FIRST ,ERRFXD
      DOUBLE PRECISION RWT,TOTWT,WEIGHT
C
C***********************************************************************
C      THIS SUBROUTINE PERFORMS TRANSFORMATIONS IF THIS OPTION IS
C         REQUESTED.
C
```

```
C
C      K                    TRANSFORMATION SET NUMBER.
C      NCON(2*K-1)           CONSTANT NUMBER TO USE.
C      NCON(2*K)             DERIVED CONSTANT.
C      NTRAN(K)              NUMBER OF TRANSFORMATION REQUESTED.
C      NXCOD(K)              VARIABLE NUMBER
C
   80 DO 500 K=1,NTRANS
      I=NCON(2*K-1)
      IF(I)100,100,110
  100 CONS=0.0
      GO TO 120
  110 CONS=CON(I)
  120 I=NXCOD(K)
      Y=X(I)
      MTRAN = NTRAN(K)
      IF(MTRAN.LE.0) MTRAN=32
  140 GO TO(150,160,170,180,190,200,210,220,230,240,250,260,270,280,290,
     X300,310,320,330,340,350,360,370,380,390,400,410,420,430,440,
     X 442,450),MTRAN
  150 CONS=Y+CONS
      GO TO 460
  160 CONS=Y*CONS
      GO TO 460
  170 CONS=CONS/Y
      GO TO 460
  180 CONS=EXP(Y)
      GO TO 460
  190 CONS=Y**CONS
      GO TO 460
  200 CONS=ALOG(Y)
      GO TO 460
  210 CONS=ALOG10(Y)
      GO TO 460
  220 CONS=SIN(Y)
      GO TO 460
  230 CONS=COS(Y)
      GO TO 460
  240 CONS=SIN(3.14159265*(CONS*Y) )
      GO TO 460
  250 CONS=COS(3.14159265*(CONS*Y) )
      GO TO 460
  260 CONS=1.0/Y
      GO TO 460
  270 CONS=EXP(CONS/Y)
      GO TO 460
  280 CONS=EXP(CONS/(Y*Y))
      GO TO 460
  290 CONS=SQRT(Y)
      GO TO 460
  300 CONS=1.0/SQRT(Y)
      GO TO 460
  310 CONS=CONS**Y
      GO TO 460
  320 CONS=10.0**Y
      GO TO 460
  330 CONS=SINH(Y)
      GO TO 460
  340 CONS=COSH(Y)
      GO TO 460
  350 CONS=(1.0-COS(Y))/2.0
      GO TO 460
```

```
  360 CONS=ATAN(Y)
      GO TO 460
  370 CONS=ATAN2(Y,CONS)
      GO TO 460
  380 CONS=Y*Y
      GO TO 460
  390 CONS=Y*Y*Y
      GO TO 460
  400 CONS=ARSIN(SQRT(Y))
      GO TO 460
  410 CONS=2.0*3.14159265*Y
      GO TO 460
  420 CONS=1.0/(2.0*3.14159265*Y)
      GO TO 460
  430 CONS=ERF(Y)
      GO TO 460
  440 CONS=GAMMA(Y)
      GO TO 460
  442 CONS=Y/CONS
      GO TO 460
  450 CONS=Y
  460 I=NCON(2*K)
  480 CON(I)=CONS
      IF(I-50) 500,500,490
  490 X(I)=CONS
  500 CONTINUE
      RETURN
      END
```

```
$IBFTC OUTPLX

C***********************************************************************
C
C***********************************************************************
C
      SUBROUTINE OUTPLT(PNCH)
      COMMON/BIG/B(60,9),SUMXY(60,9),SUMXX(1830),SUMXXI(1830)
      DOUBLE PRECISION B,SUMXY,SUMXX,SUMXXI
      COMMON/MED/BO(9),SUMX(69),SUMX2(69),SUMY2(9),ZEAN(69),
     X  CON(99),ERRMS(9),IDENT(13),IDOUT(60),NCON(200),NTERM(69),
     X  NTRAN(100),NXCOD(100),POOLED(9),REPVAR(9),RESMS(9),X(99)
      DOUBLE PRECISION BO,SUMX,SU MX2,SUMY2,ZEAN
      COMMON/SMALL/     BYPASS,BZERO,DELETE, FIRST, IFCHI,  IFSSR,
     X       IFTT,         IFWT,         INPUT,        INPUT5,      INTER,
     X       ISTRAT,       JCOL,         KONNO,        LENGTH,      LIST,
     X       NERROR,       NODEP,        NOOB,                      NOTERM,
     X       NOVAR,        NPDEG,        NRES,         NTRANS,      NWHERE,
     X       P,            PREDCT,       REPS,         RWT,
     X       STORYI,        STORYC,       STORYX,       TOTWT,       WEIGHT,
     X ERRFXD,  ECONMY,  IOUT,  ICOL
      LOGICAL ECONMY
      DOUBLE PRECISION RWT,TOTWT,WEIGHT
      LOGICAL       BYPASS,       BZERO,        DELETE,       IFCHI,
     XIFSSR,        IFTT,         IFWT,         REPS,         PREDCT,
     XSTORYC,       STORYX,       STORYI,       FIRST ,ERRFXD
      COMMON/MAX/MAXPLT
C
```

```
C*********************************************************************
      INTEGER     CELLS,      PLUS1
      DIMENSION BOUND(45), CELLBD(21),OBS(20,9), RCT(212),
     X STDERR(9),        VAR(9),              YCALC(9),          YDIFR(9),
     X Z(9)
      EQUIVALENCE (VAR,RESMS)
     0DATA  BOUND/.67448907            ,.43072720          ,.96741604
     1,.31863932 ,.67448907            ,1.15033859         ,.25334708
     2,.52439979 ,.84162001            ,1.28153777         ,.21042836
     3,.43072721 ,.67448922            ,.96741746          ,1.38298075
     4,.18001235 ,.36610621            ,.56594856          ,.79163735
     5,1.06756653,1.46521688           ,.15731067          ,.31863932
     6,.48877614 ,.67448930            ,.88714436          ,1.15034184
     7,1.53411831,.13971028            ,.28221612          ,.43072722
     8,.58945544 ,.76470731            ,.96741836          ,1.22062834
     9,1.59323335,.12566134            ,.25334701          ,.38532026
     A,.52440000 ,.67448801            ,.84161868          ,1.03642921
     B,1.28154233,1.64490172  /
C
      LOGICAL SAVRES
      DIMENSION RESPLT(1)
      DIMENSION SKEW(9),SKUR(9)
      EQUIVALENCE (RESPLT,SUMXY)
      DIMENSION ICHAR(9),MCHAR(9)
      DATA(ICHAR(I),I=1,9)/6HRESIDU,6HALS FO,6HR DEP ,6HVAR    ,1H
     X    ,6H  VS I,6HNDEP V,6HAR NO ,1H  /
      DATA(MCHAR(I),I=1,9)/6HRESIDU,6HALS FO,6HR DEP ,6HVAR    ,1H
     X  ,6H  VS P,6HREDICT,6HED VAL,2HUE /
      LOGICAL PNCH
C*********************************************************************
      JCOL=NOTERM+ NODEP
      NUVAR=NOTERM+1
      BYPASS= .FALSE.
      KOUNT= 0
      SAVRES=.FALSE.
      ITPLT=NOOB*(NWHERE+2*NODEP)
      IF(ITPLT.LE.MAXPLT) SAVRES=.TRUE.
      CALL LRLEGN(IDENT,54,0,.1,5.0,0.0)
      CALL LRLEGN(IDENT(10),24,0,.1,4.5,1.0)
C
C*********************************************************************
      IF(NOOB-20) 110,120,120
  110 BYPASS=.TRUE.
      GO TO 125
  120 CELLS=NOOB/5
      CELLS=MIN0(CELLS,20)
      I= MOD(CELLS,2)
      IF(I.NE.0) CELLS=CELLS + 1
      FCELLS= FLOAT(CELLS)
      PLUS1= CELLS + 1
      MINUS1 = CELLS -1
      NDEGCH = CELLS-3
      IR= CELLS/2-1
      IC=IR*(IR-1)/2
      IS=IR+2
      DO 122 J=1,IR
      IC=IC+1
      IBC=IS-J
      IRC=IS+J
      CELLBD(IBC)=-BOUND(IC)
      CELLBD(IRC)= BOUND(IC)
```

```
  122 CONTINUE
      CELLBD(1)=-1.0E+37
      CELLBD(PLUS1) =1.0E37
      CELLBD(IS  )=0.0
      DO 124 K=1,NODEP
      DO 124 I=1,CELLS
      OBS(I,K)=0.0
  124 CONTINUE
C
C*************************************************************************
  125 DO 130 K=1,NODEP
      SKEW(K)=0.0
      SKUR(K)=0.0
      STDERR(K)= SQRT(ERRMS(K))
  130 CONTINUE
      WRITE(LIST,135)
C
C*************************************************************************
      DO 430 J=1,NOOB
      READ(INTER) (X(I),I=1,69  ),WEIGHT
      IF(.NOT.SAVRES) GO TO 141
      INOPLT=NWHERE
      DO 140 I=1,INOPLT
      K=(I-1)*NOOB+J
  140 RESPLT(K)=X(I)
  141 CONTINUE
      DO 142 I=1,NOTERM
      K= IDOUT(I)
      X(I)= X(K)
  142 CONTINUE
      KBAR=NWHERE
      DO 143 I=1,NODEP
      IC= NOTERM+ I
      KBAR=KBAR+1
      X(IC)= X(KBAR)
  143 CONTINUE
C
C*********************************************
      DO 160 K= 1,NODEP
      YCALC(K)= BO(K)
      IF(.NOT.BZERO) YCALC(K)= 0.0
      KBAR= K+NOTERM
      DO 150 I=1,NOTERM
      YCALC(K) = YCALC(K) + B(I,K)*X(I)
  150 CONTINUE
      ACTDEV= X(KBAR)- YCALC(K)
      YDIFR(K)= ACTDEV
      Z(K)=ACTDEV/STDERR(K)
      A=ACTDEV**3
      SKEW(K)=SKEW(K)+A
      SKUR(K)=SKUR(K)+A*ACTDEV
  160 CONTINUE
      IF(.NOT.SAVRES) GO TO 179
      K=INOPLT*NOOB+J
      KBAR=K+NOOB*NODEP
      DO 175 I=1,NODEP
      ITC=(I-1)*NOOB
      ISC=K+ITC
      IS=KBAR+ITC
      RESPLT(ISC)=YCALC(I)
      RESPLT(IS)=Z(I)
  175 CONTINUE
```

```
  179 CONTINUE
      WRITE(LIST,180) (X(K),K=NUVAR,JCOL)
      WRITE(LIST,190) (YCALC(K),K=1,NODEP)
      WRITE(LIST,200) (YDIFR(K),K=1,NODEP)
      IF(PNCH) PUNCH 5250,J,(YDIFR(K),YCALC(K),K=1,NODEP)
      WRITE(LIST,210) (Z(K),K=1,NODEP)
      IF(BYPASS) GO TO 410
C
C***********************************************
      DO 250 K=1,NODEP
      DO 230 I=1,PLUS1
      IF(Z(K)-CELLBD(I)) 220,220,230
  220 OBS(I-1,K)=OBS(I-1,K)+ 1.0
      GO TO 250
  230 CONTINUE
  250 CONTINUE
C
  410 KOUNT = KOUNT +1
      IF(KOUNT.LT.10) GO TO 430
      WRITE(LIST,270) IDENT
      KOUNT=0
  430 CONTINUE
C********************************************************************
      IF(.NOT.SAVRES) GO TO 439
      ITC=NWHERE+NODEP
      DO 435 IRC=1,NODEP
      IC=(ITC+IRC-1)*NOOB+1
      DO 434 K=1,NWHERE
      IS=(K-1)*NOOB+1
      ICHAR(5)=IRC
      ICHAR(9)=K
      CALL LRCNVT(ICHAR(5),1,ICHAR(5),1,6,0)
      CALL LRCNVT(ICHAR(9),1,ICHAR(9),1,6,0)
      CALL LRTLEG(ICHAR,54)
      CALL LRPLOT(RESPLT(IS),RESPLT(IC),NOOB)
  434 CONTINUE
      IS=(NWHERE+IRC-1)*NOOB+1
      MCHAR(5)=IRC
      CALL LRCNVT(MCHAR(5),1,MCHAR(5),1,6,0)
      CALL LRTLEG(MCHAR,54)
      CALL LRPLOT(RESPLT(IS),RESPLT(IC),NOOB)
  435 CONTINUE
  439 CONTINUE
C
C***********************************************************************
      IF(BYPASS) RETURN
      DO 650 K=1,NODEP
      SKEW(K)=SKEW(K)**2/(FLOAT(NOOB)**2*ERRMS(K)**3)
      SKUR(K)=SKUR(K)/(FLOAT(NOOB)*ERRMS(K)**2)
      CHISQ=0.0
      DO 640  I=1,CELLS
      RCT(I)=OBS(I,K)
      CHISQ=CHISQ+RCT(I)**2
  640 CONTINUE
      CHISQ=FCELLS*CHISQ/FLOAT(NOOB)-FLOAT(NOOB)
      WRITE(LIST,280) NDEGCH,CHISQ,SKEW(K),SKUR(K)
      CALL HIST(K,RCT,CELLS)
  650 CONTINUE
      RETURN
  135 FORMAT( 51H FOR EACH DEPENDENT TERM AND OBSERVATION IS PRINTED
     X     /31H OBSERVED RESPONSE (Y OBSERVED)
     X     /29H CALCULATED RESPONSE (Y CALC)
```

```
   X      /28H RESIDUAL (YOBS- YCALC=YDIF)
   X      /28H STUDENTIZED  RESIDUAL ( Z ) )
 180 FORMAT(12HKY OBSERVED  ,9G13.4)
 190 FORMAT(12H Y CALC      ,9G13.4)
 200 FORMAT(12H Y DIF       ,9G13.4)
5250 FORMAT(I6,4E16.8/(6X,4E16.8))
 210 FORMAT(12H STUDENTIZED ,9G13.4)
 270 FORMAT(1H113A6,A2)
 280 FORMAT(27H1CHI-SQUARE STATISTIC WITH I6,22H DEGREES OF FREEDOM =
   X  G14.6/12H SKEWNESS = G14.6/12H KURTOSIS = G14.6)
     END
```

CRSPLT PROGRAM

CRSPLT accepts a subset of the data used for a NEWRAP problem. It can be used as a preregression analysis program to help formulate model equations to be analyzed with NEWRAP or it may be used as a postregression program by using punched output from NEWRAP to obtain more complex residual plots than direct use of NEWRAP allows.

When used as a preregression program, it will compute an X'X and C matrix including all the terms (independent and dependent) if requested. It can also compute eigenvalues and eigenvectors of the submatrix of X'X corresponding to the independent variables.

When used as a postregression program, the punched output of residuals and predicted values from NEWRAP can be plotted against new functions of the independent variables.

The input is much the same as for NEWRAP. The seven sets of input are as follows:

(1) IDENTIFICATION (I, IDENT)(I2, 13A6): IDENT is Hollerith data used to identify the problem. I indicates the number of additional cards to be read for identification (columns 1 to 78).

(2) PROBLEM SIZE (NOVAR, NODEP, NOTERM, NOOB)(3I4, I5)

NOVAR	Number of input independent variables
NODEP	Number of input dependent variables
NOTERM	Number of terms in model equation
NOOB	Number of observations

(3) TRANSFORMATIONS: This input is the same as the transformations of NEWRAP except for the upper limit of 150 transformations.

(4) FORMAT (INPUT, FMT)(I2, 13A6): INPUT specifies the unit number the input data is stored on and FMT indicates the format for reading it.

(5) PLOTTING REQUESTS

NOPLTS (I4)

(IXPLT, IYPLT)

One card supplies NOPLTS, the number of plots desired. The following cards supply pairs of integers indicating which pairs of terms to plot. The format is 40I2 (i. e., 20 plots per card). The first number of the pair (IXPLT) specifies the sequence number of the term to be used as the abscissa. The second number (IYPLT) specifies the sequence number of the term to be used as the ordinate. As an example, the following sequence of transformations and subsequent plotting requests would cause X_1^2 to be plotted against x_1 as well as against x_1^3:

MODEL SIZE	0003		
TERMS	616263		
TRANSFORMATIONS	01000061	01026162	01026263
CONSTANTS	blank card		
NOPLTS	0002		
(IXPLT, IYPLT)	0102	0302	

(6) MATRIX REQUESTS (XTXC, EIGENC)(2L1): If XTXC is F, no matrix calculations are executed. If it is T, than an X'X, X'X deviation and a correlation matrix of all the NOTERM + NODEP terms appearing on the TERMS card are computed. If EIGENC is T, then the eigenvalues and eigenvectors of the submatrix corresponding to the independent terms (the first NOTERM terms) are calculated.

(7) DATA: Same usage as in NEWRAP.

An illustrative set of input is given, followed by the corresponding sample of output and a main program listing. The subprograms TRIANG, RECT, and EIGEN are required and are the same as in NEWRAP.

```
15 SAMPLE CRSFLT PROBLEM
   DATA IS FROM  DRAPER AND SMITH
                 APPLIED REGRESSION ANALYSIS  (REFERENCE 4 OF NEWRAP REPORT)
                 CHAPTER 7
  INITIAL MODEL EQ WAS
       Y= CHAMBER PRESSURE
       X1= TEMPERATURE OF CYCLE
       X2= VIBRATION LEVEL
       X3= DROP(SHOCK)
       X4= STATIC FIRE
       Y =  B0 + B1X1 + B2X2 + B3X3 + B4X4 + ERR
```

```
 THE FOLLOWING TERMS ARE BEING CREATED FOR RESIDUAL PLOTS
1    X1                  2    X2                  3    X3                   4    X4
5    X1*X2               6    X1*X3               7    X1*X4                8    X2*X3
9    X2*X4               10    X3*X4              11    Y(PRED)             12    UNIT NO.
13    RESIDUALS
   6    1  12    24
  13
6162636465666768697071 7273
01000072020000610300006204000063050000640600007307000071010 26265012636601226467
020 26368020264690302647 0
05  (8X.F2.0,2X,4F6.0/6X,2E16.8)
  12
01130213031304130513061307130813091310131 1131213
TF
UNIT NO. 1      -75       0       0     -65    1.4
      1 -0.36550470E+01  0.50550470E+01
          2      175       0       0     150   26.3
      2 -0.76331902E+00  0.27063319E+02
          7       0       0     -65     150   29.4
      3  0.23366811E+01  0.27063319E+02
          8       0       0     165     -65    9.7
      4  0.46449530E+01  0.50550470E+01
          9       0       0       0     150   32.9
      5  0.58366811E+01  0.27063319E+02
         10     -75     -75       0     150   26.4
      6  0.50664449E+00  0.25893355E+02
         11     175     175       0     -65    8.4
      7  0.61503899E+00  0.77849610E+01
         14     -75     -75     -65     150   28.4
      8  0.25066445E+01  0.25893355E+02
         15     175     175     165     -65   11.5
      9  0.37150390E+01  0.77849610E+01
         18       0       0     -65     -65    1.3
     10 -0.37550470E+01  0.50550470E+01
         19       0       0     165     150   21.4
     11 -0.56633189E+01  0.27063319E+02
         20       0     -75     -65     -65    0.4
     12 -0.34850838E+01  0.38850838E+01
         21       0     175     165     150   22.9
     13 -0.68932328E+01  0.29793233E+02
         24       0       0       0     -65    3.7
     14 -0.13550470E+01  0.50550470E+01
          3       0     -75       0     150   26.5
     15  0.60664439E+00  0.25893355E+02
          5       0     -75       0     150   23.4
     16 -0.24933555E+01  0.25893355E+02
         16       0     -75       0     150   26.5
     17  0.60664439E+00  0.25893355E+02
          4       0     175       0     -65    5.8
     18 -0.19849610E+01  0.77849610E+01
          6       0     175       0     -65    7.4
     19 -0.38496101E+00  0.77849610E+01
         17       0     175       0     -65    5.8
     20 -0.19849610E+01  0.77849610E+01
         12       0     -75     -65    -150   28.8
     21  0.29066443E+01  0.25893355E+02
         22       0     -75     -65    -150   26.4
     22  0.50664449E+00  0.25893355E+02
         13       0     175     165     -65   11.8
     23  0.40150389E+01  0.77849610E+01
         23       0     175     165     -65   11.4
```

```
$IBFTC CRSPLX
      COMMON/B1/ X(99),CON(99),SUMX(70),SUMXX(2485),A(70,70)
     X,XDATA(12000)
      COMMON/B2/XMEAN(70),XSTD(70),SUMX2(70),NTRANS,NCON(300),
     X NTERM(70),NTRAN (150),NXCOD(150)
      DIMENSION IDENT(13),FMT(13),FMTTRI(14),CORR(1)   ,FMTSGL(3)
      DATA(FMTTRI(I),I=1,3)/6H(1H I6,6H,8G15.,2H6) /
      DATA(FMTSGL(I),I=1,3)/6H(1H I6,6H,8G15.,2H6) /
      LOGICAL XTXC,EIGENC
      EQUIVALENCE (CORR,A)
      DATA ICHAR(2)/6H  VS   /
      COMMON IXPLT(400),IYPLT(400)
      DIMENSION  ICHAR(3)
C*********************************************************************
   10 READ(5,110) I,IDENT
      WRITE(6,111)  IDENT
      DO 100 J=1,19863
      X(J)=0.0
  100 CONTINUE
  113 IF(I)120,120,115
  115 READ(5,300) FMT
      WRITE(6,301) FMT
      I=I-1
      GO TO 113
  120 READ(5,112) NOVAR,NODEP,NOTERM,NOOB
      WRITE(6,305)NOVAR,NODEP,NOTERM,NOOB
      MVTERM=NOTERM+NODEP
      L=MVTERM*NOOB
      IF(L.GT.12000) GO TO 1000
      READ (5,282) NTRANS,KONNO
      IF(NTRANS)255,255,220
  220 READ(5,230)(NTERM(K),K=1,MVTERM)
      WRITE(6,235)(NTERM(K),K=1,MVTERM)
      READ(5,230)(NXCOD(I),NTRAN(I),NCON(2*I-1),NCON(2*I),I=1,NTRANS)
      WRITE(6,240)(NXCOD(I),NTRAN(I),NCON(2*I-1),NCON(2*I),I=1,NTRANS)
      IF(KONNO) 255,255,250
  250 READ(5,260)(CON(I),I=1,KONNO)
      WRITE(6,262)(CON(I),I=1,KONNO)
  255 READ(5,3000)       IUIN,FMT
      WRITE(6,3001)     IUIN,FMT
      READ(5,282) NOPLTS
      IF(NOPLTS.LE.0) GO TO 320
      IF(NOPLTS.GT.300) GO TO 2000
      READ(5,230)     (IXPLT(I),IYPLT(I),I=1,NOPLTS)
      WRITE(6,5000) (IXPLT(I),IYPLT(I),I=1,NOPLTS)
  320 CONTINUE
      READ(5,4000) XTXC,EIGENC
      NDRD=NOVAR+NODEP
C
C*******************************************************************
      RNOOB = 1.0   /FLOAT(NOOB)
      DO 690   J=1,NOOB
      READ(IUIN,FMT) (X(I),I=1,NDRD)
      IF(NTRANS)450,450,340
  340 CALL TRANS
      DO 430 K=1,MVTERM
      I= NTERM(K)
      X(K)=CON(I)
  430 CONTINUE
```

```
450 CONTINUE
    IF(.NOT.XTXC) GO TO 620
    IR=0
    DO 610 I=1,MVTERM
    SUMX(I) =SUMX(I) + X(I)
    DO 600 II=1,I
    IR=IR+1
    SUMXX(IR)= SUMXX(IR) + X(I)*X(II)
600 CONTINUE
610 CONTINUE
620 CONTINUE
    IA =J-NOOB
    DO 650 I=1,MVTERM
    IX = IA + I*NOOB
    XDATA(IX)=X(I)
650 CONTINUE
690 CONTINUE
    IF(.NOT.XTXC) GO TO 720
    WRITE(6,1010)
    CALL RECT(NOOB,MVTERM,NOOB,MVTERM,XDATA,FMTSGL)
    WRITE(6,1020)
     CALL TRIANG(SUMXX,MVTERM,8,FMTTRI)
    IR= 0
     DO 700 I=1,MVTERM
    IR= IR+I
    SUMX2(I)= SUMXX(IR)  -SUMX(I)**2*RNOOB
700 XMEAN(I)= SUMX(I)*RNOOB
    IR= 1
    DO 710 J=1,MVTERM
    DO 710  K=1,J
    SUMXX(IR)= SUMXX(IR) - SUMX(K)*SUMX(J)*RNOOB
    CORR(IR) = SUMXX(IR)/ SQRT(SUMX2(J)*SUMX2(K))
    IR=IR+1
710 CONTINUE
    WRITE(6,1025)
     CALL TRIANG(SUMXX,MVTERM,8,FMTTRI)
    WRITE(6,1030)
    CALL TRIANG(CORR,MVTERM,8,FMTTRI)
    IF(.NOT.EIGENC) GO TO 720
    CALL  EIGEN(SUMXX,A,NOTERM,0)
    WRITE(6,1040)
    J= 0
    DO  718 I=1,NOTERM
    J= J+ I
718 SUMXX(I)=SUMXX(J)
    WRITE(6,1050)(SUMXX(I),I=1,NOTERM)
    WRITE(6,1060)
    CALL RECT(NOTERM,NOTERM,NOTERM,NOTERM,A,FMTTRI)
720 CONTINUE
    CALL LRLEGN(IDENT,54,0,.1,5.0,0.0)
    CALL LRLEGN(IDENT(10),24,0,.1,4.5,1.0)
    DO 900 K=1,NOPLTS
    ICHAR(1)=IXPLT(K)
    ICHAR(3)=IYPLT(K)
    IS1=(IXPLT(K)-1)*NOOB+1
    IS2=(IYPLT(K)-1)*NOOB+1
    CALL LRCNVT(ICHAR(1),1,ICHAR(1),1,6,0)
    CALL LRCNVT(ICHAR(3),1,ICHAR(3),1,6,0)
    CALL LRTLEG(ICHAR,18)
    CALL LRPLOT(XDATA(IS1),XDATA(IS2),NOOB)
900 CONTINUE
    GO TO 10
```

```
1000 WRITE(6,95)L
     STOP
  95 FORMAT(62H THE REQUIRED NUMBER OF LOCATIONS EXCEEDS THE 12000 AVAI
    XLABLE  I8)
2000 WRITE(6,2005) NOPLTS
     STOP
2005 FORMAT(25H MAX NO. OF PLOTS IS 300   I8)
 110 FORMAT(I2,13A6)
 111 FORMAT(1H1,13A6)
 112 FORMAT(3I4,I5)
4000 FORMAT(2L1)
5000 FORMAT(1HK/(15(1X,I2,1X,I2,2X)))
 300 FORMAT(13A6,A2)
 301 FORMAT(1H 13A6,A2)
 305 FORMAT(1H  3I6)
 282 FORMAT( 2I4)
 230 FORMAT(40I2)
 235 FORMAT(11H TERMS ARE  / (1H 30I4))
 240 FORMAT(25H THE TRANSFORMATIONS ARE   /(1H 5(4I4,5X)))
 260 FORMAT( 5E15.7)
 262 FORMAT(19H THE CONSTANTS ARE   /((1H  8G15.7)))
1010 FORMAT(16H1THE DATA MATRIX  )
1020 FORMAT(26H2THE X TRANSPOSE X MATRIX )
1025 FORMAT(32H X TRANSPOSE X DEVIATIONS MATRIX  )
1030 FORMAT(23H2THE CORRELATION MATRIX )
1040 FORMAT(46H2FOLLOWING ARE EIGENVALUES OF X TRANS X MATRIX)
1050 FORMAT(1H 8G16.7)
1060 FORMAT(53HKEIGENVECTORS BY COLUMNS IN SAME ORDER AS EIGENVALUES)
3000 FORMAT(I2,13A6)
3001 FORMAT(1H  I6,2X,13A6)
     END
```

```
$IBFTC TRANSX

      SUBROUTINE TRANS
C*********************************************************************
C
      COMMON/B1/ X(99),CON(99),SUMX(70),SUMXX(2485),A(70,70)
     X,XDATA(12000)
      COMMON/B2/XMEAN(70),XSTD(70),SUMX2(70),NTRANS,NCON(300),
     X NTERM(70),NTRAN (150),NXCOD(150)
C
C*********************************************************************
C     THIS SUBROUTINE PERFORMS TRANSFORMATIONS IF THIS OPTION IS
C        REQUESTED.
C
C
C     K                   TRANSFORMATION SET NUMBER.
C     NCON(2*K-1)          CONSTANT NUMBER TO USE.
C     NCON(2*K)            DERIVED CONSTANT.
C     NTRAN(K)             NUMBER OF TRANSFORMATION REQUESTED.
C     NXCOD(K)             VARIABLE NUMBER
C
  80 DO 500 K=1,NTRANS
     I=NCON(2*K-1)
     IF(I)100,100,110
```

```
100  CONS=0.0
     GO TO 120
110  CONS=CON(I)
120  I=NXCOD(K)
     Y=X(I)
     MTRAN = NTRAN(K)
     IF(MTRAN.LE.0) MTRAN=32
140  GO TO(150,160,170,180,190,200,210,220,230,240,250,260,270,280,290,
    X300,310,320,330,340,350,360,370,380,390,400,410,420,430,440,
    X 442,450 ),MTRAN
150  CONS=Y+CONS
     GO TO 460
160  CONS=Y*CONS
     GO TO 460
170  CONS=CONS/Y
     GO TO 460
180  CONS=EXP(Y)
     GO TO 460
190  CONS=Y**CONS
     GO TO 460
200  CONS=ALOG(Y)
     GO TO 460
210  CONS=ALOG10(Y)
     GO TO 460
220  CONS=SIN(Y)
     GO TO 460
230  CONS=COS(Y)
     GO TO 460
240  CONS=SIN(3.14159265*(CONS*Y) )
     GO TO 460
250  CONS=COS(3.14159265*(CONS*Y) )
     GO TO 460
260  CONS=1.0/Y
     GO TO 460
270  CONS=EXP(CONS/Y)
     GO TO 460
280  CONS=EXP(CONS/(Y*Y))
     GO TO 460
290  CONS=SQRT(Y)
     GO TO 460
300  CONS=1.0/SQRT(Y)
     GO TO 460
310  CONS=CONS**Y
     GO TO 460
320  CONS=10.0**Y
     GO TO 460
330  CONS=SINH(Y)
     GO TO 460
340  CONS=COSH(Y)
     GO TO 460
350  CONS=(1.0-COS(Y))/2.0
     GO TO 460
360  CONS=ATAN(Y)
     GO TO 460
370  CONS=ATAN2(Y/CONS)
     GO TO 460
380  CONS=Y*Y
     GO TO 460
390  CONS=Y*Y*Y
     GO TO 460
400  CONS=ARSIN(SQRT(Y))
     GO TO 460
```

```
410 CONS=2.0*3.14159265*Y
    GO TO 460
420 CONS=1.0/(2.0*3.14159265*Y)
    GO TO 460
430 CONS=ERF(Y)
    GO TO 460
440 CONS=GAMMA(Y)
    GO TO 460
442 CONS=Y/CONS
    GO TO 460
450 CONS=Y
460 I=NCON(2*K)
    IF(I)470,470,480
470 CON(K)=CONS
    GO TO 500
480 CON(I)=CONS
    IF(I-60) 500,500,490
490 X(I)=CONS
500 CONTINUE
    RETURN
    END
```

```
 SAMPLE CRSPLT PROBLEM
   DATA IS FROM  DRAPER AND SMITH
                 APPLIED REGRESSION ANALYSIS  (REFERENCE 4 OF NEWRAP REPORT)
                 CHAPTER 7
  INITIAL MODEL EQ WAS
        Y= CHAMBER PRESSURE
        X1= TEMPERATURE OF CYCLE
        X2= VIBRATION LEVEL
        X3= DROP(SHOCK)
        X4= STATIC FIRE
        Y =  B0 + B1X1 + B2X2 + B3X3 + B4X4 + ERR
 THE FOLLOWING TERMS ARE BEING CREATED FOR RESIDUAL PLOTS
1   X1              2   X2              3   X3              4   X4
5   X1*X2           6   X1*X3           7   X1*X4           8   X2*X3
9   X2*X4           10   X3*X4          11   Y(PRED)        12   UNIT NO.
13   RESIDUALS
    6     1     12
   24
TERMS ARE
  61  62  63  64  65  66  67  68  69  70  71  72  73
THE TRANSFORMATIONS ARE
   1   0   0  72          2   0   0  61          3   0   0  62          4   0   0  63          5   0   0  64
   6   0   0  73          7   0   0  71          1   2  62  65          1   2  63  66          1   2  64  67
   2   2  63  68          2   2  64  69          3   2  64  70
     5    (8X,F2.0,2X,4F6.0/6X,2E16.8)

 1 13     2 13     3 13     4 13     5 13     6 13     7 13     8 13     9 13    10 13    11 13    12 13
```

THE DATA MATRIX

1	-75.00000	0	0	-65.00000	0	0	-65.00000	-0	
2	175.0000	0	0	150.0000	0	0	300.0000	0	
3	0	0	-65.00000	150.0000	0	-455.0000	[illegible]	-0	
4	0	0	165.0000	-65.00000	0	1320.000	-520.0000	0	
5	0	0	0	150.0000	0	0	1350.000	0	
6	-75.00000	-75.00000	0	150.0000	-750.0000	0	1500.000	-0	
7	175.0000	175.0000	0	-65.00000	1925.000	0	-715.0000	0	
8	-75.00000	-75.00000	-65.00000	150.0000	-1050.000	-910.0000	2100.000	4875.000	
9	175.0000	175.0000	165.0000	-65.00000	2625.000	2475.000	-975.0000	28875.00	
10	0	0	-65.00000	-65.00000	0	-1170.000	-1170.000	-0	
11	0	0	165.0000	150.0000	0	3135.000	2850.000	0	
12	0	-75.00000	-65.00000	-65.00000	-1500.000	-1300.000	-1300.000	-0	
13	0	175.0000	165.0000	150.0000	3675.000	3465.000	3150.000	0	
14	0	0	0	-65.00000	0	0	-1560.000	0	
15	0	-75.00000	0	150.0000	-225.0000	0	450.0000	0	
16	0	-75.00000	0	150.0000	-375.0000	0	750.0000	0	
17	0	-75.00000	0	150.0000	-1200.000	0	2400.000	0	
18	0	175.0000	0	-65.00000	700.0000	0	-260.0000	0	
19	0	175.0000	0	-65.00000	1050.000	0	-390.0000	0	
20	0	175.0000	0	-65.00000	2975.000	0	[illegible]	0	
21	0	-75.00000	-65.00000	-150.0000	-900.0000	-780.0000	-1800.000	-0	
22	0	-75.00000	-65.00000	-150.0000	-1650.000	-1430.000	-3300.000	-0	
23	0	175.0000	165.0000	-65.00000	2275.000	2145.000	-845.0000	0	
24	0	175.0000	165.0000	-65.00000	4025.000	3795.000	-1495.000	0	

1	4875.000	-0	5.055047	1.000000	-3.655047
2	26250.00	0	27.06332	2.000000	-0.763319
3	0	0	27.06332	7.000000	2.336681
4	-0	-0	5.055047	8.000000	4.644953
5	0	0	27.06332	9.000000	5.836681
6	-11250.00	-11250.00	25.89335	10.00000	0.576644
7	-11375.00	-11375.00	7.784961	11.00000	0.615039
8	-11250.00	-11250.00	25.89335	14.00000	2.576644

9	-11375.00	-11375.00	7.784961	15.00000	3.715039
10	-0	-0	5.055047	18.00000	-3.755047
11	0	0	27.06332	19.00000	-5.663319
12	-0	4875.000	3.885084	20.00000	-3.485084
13	0	26250.00	29.79323	21.00000	-6.893233
14	-0	-0	5.055047	24.00000	-1.355047
15	0	-11250.00	25.89335	3.000000	0.606644
16	0	-11250.00	25.89335	5.000000	-2.493356
17	0	-11250.00	25.89335	16.00000	0.606644
18	-0	-11375.00	7.784961	4.000000	-1.984961
19	-0	-11375.00	7.784961	6.000000	-0.384961
20	-0	-11375.00	7.784961	17.00000	-1.984961
21	-0	11250.00	25.89335	12.00000	2.906644
22	-0	11250.00	25.89335	22.00000	0.506644
23	-0	-11375.00	7.784961	13.00000	4.015039
24	-0	-11375.00	7.784961	23.00000	3.615039

THE X TRANSPOSE X MATRIX

1	108750.0							
2	72500.00	290000.0						
3	33750.00	135000.0	188700.0					
4	-14125.00	-82250.00	15050.00	320700.0				
5	931250.0	0.394250E+07	0.241050E+07	-521125.0	0.650000E+08			
6	501375.0	0.241050E+07	0.308820E+07	644525.0	0.453525E+08	0.554528E+08		
7	-508375.0	-521125.0	644525.0	0.382600E+07	-0.175562E+07	0.160571E+08	0.599937E+08	
8	0.468750E+07	0.468750E+07	0.444750E+07	-0.114562E+07	0.706781E+08	0.670294E+08	-0.179156E+08	0.857521E+09
9	0.193437E+07	-0.229375E+07	-0.114562E+07	0.172438E+07	-0.315062E+08	-0.179156E+08	-0.137181E+08	-0.383297E+09
10	-0.229375E+07	-0.717500E+07	-0.234750E+07	-0.301625E+07	-0.761969E+08	-0.257269E+08	0.377937E+07	-0.383297E+09
11	3197.685	865.0004	6679.541	27249.98	65569.14	156194.1	221582.0	351020.9
12	3025.000	11600.00	10290.00	400.0000	198500.0	191380.0	-44650.00	501375.0
13	672.3147	-0.152588E-04	500.4591	-1023.986	423.3643	-1394.072	-29910.56	119491.6

9	0.122473E+10				
10	0.511906E+09	0.250450E+10			
11	-24652.88	-692766.1	9052.137		
12	-508375.0	-521125.0	4778.626	4900.000	
13	-121009.6	-265546.4	0.686646E-04	-82.22644	256.3119

X TRANSPOSE X DEVIATIONS MATRIX

1	105000.0							
2	62500.00	263333.3						
3	26250.00	115000.0	173700.0					
4	-19375.00	-96250.00	4550.000	313350.0				
5	786250.0	0.355583E+07	0.212050E+07	-724125.0	0.593933E+08			
6	372750.0	0.206750E+07	0.283095E+07	464450.0	0.403790E+08	0.510410E+08		
7	-513375.0	-534458.3	634525.0	0.381900E+07	-0.194896E+07	0.158856E+08	0.599871E+08	
8	0.426562E+07	0.356250E+07	0.360375E+07	-0.173625E+07	0.543656E+08	0.525591E+08	-0.184781E+08	0.81[illegible]E+09
9	0.211094E+07	-0.182292E+07	-792500.0	0.197156E+07	-0.246792E+08	-0.118595E+08	-0.134827E+08	-0.363434E+09
10	-0.126563E+07	-0.443333E+07	-291250.0	-0.157688E+07	-0.364427E+08	0.953781E+07	0.515021E+07	-0.267633E+09
11	-1776.064	-12398.33	-3267.958	20286.74	-126649.2	-14405.53	224950.4	-208526.0
12	-725.0000	1600.000	2790.000	-4850.000	53500.00	62755.00	-49550.00	79500.00
13	672.3146	-0.140429E-03	500.4590	-1023.986	423.3624	-1394.073	-29910.56	119491.6

9	0.121642E+10				
10	0.463499E+09	0.222262E+10			
11	209527.8	670870.2	2455.289		
12	-331812.5	507000.0	-195.1232	1150.000	
13	-121009.6	-265546.3	0.640825E-05	-82.22649	256.3119

THE CORRELATION MATRIX

1	1.000000							
2	0.375865	1.000000						
3	0.194372	0.537706	1.000000					
4	-0.106815	-0.335068	0.195028E-01	1.000000				
5	0.314845	0.899124	0.660191	-0.167853	1.000000			
6	0.161014	0.563940	0.950764	0.116135	0.733277	1.000000		
7	-0.204556	-0.134472	0.196571	0.880858	-0.326516E-01	0.287088	1.000000	
8	0.462516	0.243916	0.303804	-0.108977	0.247853	0.258480	-0.938229E-01	1.000000
9	0.186784	-0.101853	-0.545202E-01	0.100984	-0.918163E-01	-0.475955E-01	-0.499121E-01	-0.366119
10	-0.828471E-01	-0.183250	-0.148229E-01	-0.597516E-01	-0.100302	0.283176E-01	0.141[illegible]47E-01	-0.109455
11	-0.110615	-0.487595	-0.158243	0.731385	-0.331652	-0.406929E-01	0.586147	-0.147859
12	-0.659773E-01	0.919429E-01	0.197404	-0.255492	0.204709	0.259024	-0.189035	0.824677E-01
13	0.129597	-0.170930E-07	0.752040E-01	-0.114260	0.343130E-02	-0.121883E-01	-0.241219	0.[illegible]62236

9	1.000000				
10	0.281886	1.000000			
11	0.121241	0.287180	1.000000		
12	-0.280545	0.317122	-0.116120	1.000000	
13	-0.216717	-0.351822	0.807800E-08	-0.151453	1.000000

01 UNIT05. EOF. REC= [illegible] FIL= [illegible]2

CREDUC PROGRAM

In optimum-seeking experimentation involving many independent variables, quadratic response surfaces are often used. A development of the most important aspect of the design and analysis of response surface experiments can be found in Davies (ref. 9) and Box and Hunter (ref. 10). A discussion of the interpretation of a quadratic surface fitted to a large experiment is given in reference 11.

The general form of a quadratic surface is given by

$$
\begin{aligned}
y &= b_0 + b'X + X'BX + \epsilon \\
&= b_0 + (b_1, b_2, \ldots, b_p)\begin{pmatrix} X_1 \\ \cdot \\ \cdot \\ \cdot \\ X_p \end{pmatrix} \\
&\quad + (X_1, \ldots, X_p)\begin{pmatrix} b_{11} & \frac{1}{2}b_{12} & \cdots & \frac{1}{2}b_{1p} \\ \frac{1}{2}b_{12} & b_{22} & \cdots & \frac{1}{2}b_{2p} \\ \vdots & & \ddots & \\ \cdot & & & b_{pp} \end{pmatrix}\begin{pmatrix} X_1 \\ \cdot \\ \cdot \\ \cdot \\ X_p \end{pmatrix} + \epsilon
\end{aligned} \tag{26}
$$

The analysis of such an equation is simplified by two calculations: (1) the calculation of the stationary point of the surface and (2) a transformation of axes to new independent variables which changes the B matrix to a diagonal matrix.

The stationary point of the surface is the solution X_s to

$$\frac{\partial y}{\partial x_i} = 0 \qquad i = 1, p$$

The transformation of the axes is given by the computation of the orthogonal matrix P which reduces B to a diagonal matrix; that is,

$$P'BP = \begin{pmatrix} \lambda_1 & & & 0 \\ & \cdot & & \\ & & \cdot & \\ & & & \cdot \\ 0 & & & \lambda_p \end{pmatrix}$$

The λ_i are the eigenvalues of B. The new variables are given by

$$Z = P'X$$

where P is the matrix whose columns are the eigenvectors of B. If two successive transformations of the variables are made as follows:

$$W = X - X_s$$

$$Z = P'W$$

then equation (26) becomes

$$y - y_s = \lambda_1 Z_1^2 + \lambda_2 Z_2^2 + \ldots + \lambda_p Z_p^2 \tag{27}$$

where

$$y_s = b_0 + b'X_s + X_s'BX_s \tag{28}$$

From examination of equation (27), some general conclusions can be drawn concerning the attainment of a maximized response. For example, consider just two of the possible results.

(1) Suppose all the $\lambda_i \leq 0$ and X_s is near or in the region of X at which the experiments were performed. Then clearly any deviation of Z from $Z = 0$ will decrease the response. Thus $Z = 0$ (or equivalently $X = X_s$) is a maximum and is the combination of independent variables the experimenter seeks.

(2) Some $\lambda_i < 0$ and some $\lambda_i > 0$, and X_s is close to the region of experimentation. Thus X_s represents what is sometimes called a saddlepoint. Moving in some directions will cause a decrease in y and moving in other directions will cause an increase in y. Thus the experimenter could move from X_s in the direction that corresponds to the direction of the Z which has the largest positive coefficient in equation (27). This will increase y most rapidly from the value of y_s.

FORTRAN STATEMENT

```
  SAMPLE CANONICAL REDUCTION PROBLEM
   5
 (5F12.0)
147.2686
-8.989120   -6.817975   -14.60964   9.248688    16.19698
1.805520
-2.126719   -1.475730
-7.314218   -16.10219   2.274270
1.310780    -2.477188   -1.289688   2.236770
2.664464    1.827434    2.389934    6.014934    -12.58198
```

NASA-C-836 (REV 9-14-59)

Figure 8. - Sample CREDUC input.

The INPUT (Sample input is shown in fig. 8) is as follows:

(1) Identification: One card, all 80 columns.

(2) The number of independent variations JFAC. (I4) (JFAC must be less than or equal to 15)

(3) Variable Format. One card, all 80 columns.

(4) b_0 (according to FORMAT in item (3)).

(5) $b_1, \ldots, b_{JFAC}$ (according to format in item (3)).

(6)

$$
\begin{array}{llll}
b_{1,1} & & & \\
b_{1,2} & b_{2,2} & & \\
b_{1,3} & b_{2,3} & b_{3,3} & \\
\cdot & & & \\
\cdot & & & \\
\cdot & & & \\
b_{1,JFAC} & b_{2,JFAC} & \cdots & b_{JFAC,JFAC}
\end{array}
$$

One READ statement for each line, according to the format in item (3). As an example, consider the following estimated equation:

$$y = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + b_4X_4$$

$$\begin{array}{lllll} +b_{11}X_1^2 & & & & \\ +b_{12}X_1X_2 & +b_{22}X_2^2 & & & \\ +b_{13}X_1X_3 & +b_{23}X_2X_3 & +b_{33}X_3^2 & & \\ +b_{14}X_1X_4 & +b_{24}X_2X_4 & +b_{34}X_3X_4 & +b_{44}X_4^2 & \\ +b_{15}X_1X_5 & +b_{25}X_2X_5 & +b_{35}X_3X_5 & +b_{45}X_4X_5 & +b_{55}X_5^2 \end{array}$$

$$y = 147.2686 - 8.989120\ X_1 - 6.817975\ X_2 - 14.60964\ X_3 + 9.248688\ X_4 + 14.19698\ X_5$$

$$\begin{array}{lllll} +1.805520\ X_1^2 & & & & \\ -2.126719\ X_1X_2 & -1.475730\ X_2^2 & & & \\ -7.314218\ X_1X_3 & -16.10219\ X_2X_3 & +2.274270\ X_3^2 & & \\ +1.310780\ X_1X_4 & -2.477188\ X_2X_4 & -1.289688\ X_3X_4 & +2.236770\ X_4^2 & \\ +2.664464\ X_1X_5 & +1.827434\ X_2X_5 & +2.389934\ X_3X_5 & +6.014934\ X_4X_5 & -12.58198\ X_5^2 \end{array}$$

A canonical reduction of this is given in the sample output. The listing of the main program is supplied. The subprograms TRIANG, DGELG, EIGEN, and RECT are required. TRIANG and RECT are as in NEWRAP. DGELG and EIGEN are the double precision general linear equation and eigenvalue routines from reference 12.

```
$IBFTC CRFDJC

      DIMENSION BL(15), BS(105),BLSAVE(15),BSAVE(105),FMT(14),XIN(225)
      DOUBLE PRECISION BL,BS,BLSAVE,BSAVE,XIN,BZERO,YS
      DIMENSION SBL(15),SBS(225)      ,FMTTRI(14)
      DATA(FMTTRI(I),I=1,4)/6H(5H RO,6HW I5,2,6HX,(8G1,6H5.6))  /
  999 READ (5,1001) FMT
      WRITE(5,1002) FMT
      READ(5,1003) JFAC
      WRITE(6.1004) JFAC
```

```
      READ(5,1001) FMT                                                  8
      READ(5,FMT) BZERO                                                 10
      WRITE(6,1005) BZERO                                               11
      READ(5,FMT) (BL(I),I=1,JFAC)                                      12
      IE=0
      DO 5 I=1,JFAC
      IE=IE+I
      IS=IE-I+1
      READ(5,FMT) (BS(K),K=IS,IE)                                       24
      DO 4 K= IS,IE
      BSAVE(K)=BS(K)
    4 SBS(K)= SNGL(BS(K))
      BLSAVE(I)=BL(I)
      SBL(I)= SNGL(BL(I))
      BL(I)=-BL(I)
    5 CONTINUE
      LENGTH = JFAC*(JFAC+1)/2
      WRITE(6,1006)                                                     46
      WRITE(6,1007)(SBL(I),I=1,JFAC)                                    47
      WRITE(6,1008)                                                     54
      CALL TRIANG(SBS,JFAC,8,FMTTRI)                                    55
      IJ=1
      XIN(1)= BS(1)*2.0D0
      DO 50 I=2,JFAC
      I1= I-1
      IIK=I
      IIJ = JFAC*I1
      DO 40 J=1,I1
      IJ= IJ+1
      BSAVE(IJ)=0.50D0*BS(IJ)
      XIN(IIK)=BS(IJ)
      IIK=IIK+JFAC
      IIJ= IIJ+1
      XIN(IIJ)= BS(IJ)
   40 CONTINUE
      IIJ=IIJ+1
      IJ= IJ+1
      XIN(IIJ)=2.0D0*BS(IJ)
   50 CONTINUE
      EPS=1.0E-10
      CALL DGELG(BL,XIN,JFAC,1,EPS,IER)                                 84
      IF(IER.NE.0) WRITE(6,1009) IER                                    86
      WRITE(6,1010)                                                     87
      WRITE(6,1007) (BL(I),I=1,JFAC)                                    88
      IJ=0
      YS= BZERO
      DO 150 I=1,JFAC
      YS=YS+BL(I)*BLSAVE(I)
      DO 140 J=1,I
      IJ= IJ+1
  140 YS= YS+ BL(I)*BL(J)*BS    (IJ)
  150 CONTINUE
      WRITE(5,1011) YS
C                                                                       110
      IJ=0
      DO 160 L=1,LENGTH
      IJ=IJ+1
  160 SBS(IJ)=SNGL(BSAVE(IJ))
      CALL EIGEN(BSAVE,XIN,JFAC,0)                                      120
      IJ=0
      DO 200 I=1,JFAC
      IJ=IJ+I
  200 BL(I)= BSAVE(IJ)
      WRITE(6,1012)                                                     130
      WRITE(6,1007) (BL(I),I=1,JFAC)                                    131
      WRITE(6,1013)                                                     138
      JJ=JFAC*JFAC
      DO 210 I=1,JJ
```

```
  210 SBS(I)= SNGL(XIN(I))
      CALL RECT(JFAC,JFAC,JFAC,JFAC,SBS,FMTTRI)                                148
      GO TO 999
C***  ****************************************************************
C
C
 1001 FORMAT(13A6,A2)
 1002 FORMAT(1H1 13A6,A2)
 1003 FORMAT(I4)
 1004 FORMAT(1H I4)
 1005 FORMAT(24HKTHE COEFFS ARE BZERO    G16.8)
 1006 FORMAT(24HK                  LINEAR    )
 1007 FORMAT(1H 8G16.8)
 1008 FORMAT(24HK            SECOND ORDER    )
 1009 FORMAT(51HKTHE SOLUTION FOR STATIONARY POINT MAY BE INCORRECT /
     X  6H IER=  I3)
 1010 FORMAT(26HKTHE STATIONARY POINT IS    )
 1011 FORMAT(20HKTHE VALUE OF YS IS  G16.8)
 1012 FORMAT(14HKEIGENVALUES  )
 1013 FORMAT(14HKEIGENVECTORS )
      END
```

```
  SAMPLE CANONICAL REDUCTION PROBLEM
   5

THE COEFFS ARE BZERO      147.268600

                 LINEAR
 -8.98912001     -6.81797498      -14.6096400       9.24868798       16.1969800

             SECOND ORDER
ROW     1      1.805520
ROW     2     -2.126719      -1.475730
ROW     3     -7.314218      -16.10219       2.274270
ROW     4      1.310780      -2.477188      -1.289688       2.236770
ROW     5      2.664464       1.827434       2.389934       6.014934      -12.58198

THE STATIONARY POINT IS
  3.05811524      -1.96462563       0.12068067      -3.89057544      -0.93711068E-01

THE VALUE OF YS IS    120.589275

EIGENVALUES
  9.35986030       3.75144443       1.20220299      -8.09779024      -13.9568675

EIGENVECTORS
ROW     1     -0.300108       0.519711       0.750815       0.238709      -0.138316
ROW     2     -0.549544      -0.275150      -0.293631       0.690293      -0.244071
ROW     3      0.779700       0.809764E-02   0.804860E-01   0.577909      -0.227038
ROW     4     -0.219489E-02   0.790049      -0.582318      -0.228827E-02  -0.191618
ROW     5      0.105615E-02   0.173062      -0.669725E-01   0.364045       0.912707
```

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, January 25, 1972,
132-80.

APPENDIX - BORROWED ROUTINES

Some of the routines used in the programs were taken from the literature. Both INVXTX and TRIANG are by Webb and Galley (ref. 13), and EIGEN and HIST are from the IBM programmer's manual (ref. 12), as are DGELG and the double precision version of EIGEN.

Listing of INVXTX and TRIANG are given here, as follows:

```
$IBFTC TRIANX
      SUBROUTINE TRIANG(A,B,NN,NKOL,FORMAT,II)
      DIMENSION FORMAT(1)
      DIMENSION A(1)    ,B(1)
      DOUBLE PRECISION B
    1 FORMAT (1H1)
    3 FORMAT(1H /1H /1H  )
      COMMON/SMALL/DUM(15),LIST
      N = NN
      NCOL = NKOL
      KLUMPS = N/NCOL
C
      KEEPTR = 0
      K1 = 1
      K2 = NCOL - 1
      K3 = NCOL
      IF (KLUMPS .EQ. 0)  GO TO 120
C
      DO 90 KLUMP=1,KLUMPS
      ITR1 = KEEPTR
      I = -1
      ILO = (KLUMP-1)*NCOL + ITR1 + 1
      DO 30 K=K1,K2
      I = I + 1
      ITR1 = ITR1 + K - 1
      ILO  = ILO  + K - 1
      IHI = ILO + I
      GO TO (26,28),II
   26 WRITE(LIST,FORMAT) K,(A(J),J=ILO,IHI)
      GO TO 30
   28 WRITE(LIST,FORMAT) K,(B(J),J=ILO,IHI)
   30 CONTINUE
      KEEPTR = ITR1 + K2
      DO 60 K=K3,N
      ITR1 = ITR1 + K - 1
      ILO  = ILO  + K - 1
      IHI = ILO + NCOL - 1
      GO TO(56,58),II
   56 WRITE(LIST,FORMAT) K,(A(J),J=ILO,IHI)
      GO TO 60
   58 WRITE(LIST,FORMAT) K,(B(J),J=ILO,IHI)
   60 CONTINUE
      K1 = K1 + NCOL
      K2 = K2 + NCOL
      K3 = K3 + NCOL
   90 WRITE(LIST,3)
C
```

```
  120 ITR1 = KEEPTR
      IF (K1 .GT. N)  GO TO 180
      I = -1
      ILO = KLUMPS*NCOL + ITR1 + 1
      DO 150 K=K1,N
      I = I + 1
      ITR1 = ITR1 + K - 1
      ILO  = ILO  + K - 1
      IHI = ILO + I
      GO TO (146,148),II
  146 WRITE(LIST,FORMAT) K,(A(J),J=ILO,IHI)
      GO TO 150
  148 WRITE(LIST,FORMAT) K,(B(J),J=ILO,IHI)
  150 CONTINUE
C
  180 RETURN
      END
```

```
$IBFTC INVXXX

      SUBROUTINE INVXTX(A, NN, D, FACT)
C
C     ASSUMES THE MATRIX A IS SYMMETRIC AND POSITIVE DEFINITE, AND ONLY
C     THE  UPPER TRIANGLE IS STORED AS A ONE-DIMENSIONAL ARRAY IN THE
C     ORDER A(1,1), A(1,2), A(2,2), A(1,3), A(2,3), A(3,3), ..., A(N,N).
C     NN IS THE ORDER N OF THE INPUT MATRIX A.
C     D  IS (ON EXIT) THE DETERMINANT OF A, DIVIDED BY FACTOR**NN.
C
      DIMENSION A(1)
      DOUBLE PRECISION A,PV,F
      N = NN
      ITR1 = 0
      DO 145 K=1,N
C
      ITR1 = ITR1+K-1
      KP1 = K+1
      KM1 = K-1
      KK = ITR1+K
      PV =  1.0D0/A(KK)
C
      ITR2 = 0
      IF (K-1) 150,80,50
C
C     REDUCE TOP PART OF TRIANGLE, LEFT OF PIVOTAL COLUMN
   50 DO 60 J=1,KM1
      ITR2 = ITR2+J-1
      KJ = ITR1+J
      F = A(KJ)*PV
      DO 60 I=1,J
      IJ = ITR2+I
      IK = ITR1 + I
   60 A(IJ) = A(IJ) + A(IK)*F
C
      IF (K-N) 70,120,150
C
C     REDUCE REST OF TRIANGLE, RIGHT OF PIVOTAL COLUMN
   70 ITR2 = ITR1
```

```
   80 DO 110 J=KP1,N
      ITR3 = ITR1
      ITR2 = ITR2+J-1
      KJ = ITR2+K
      F = A(KJ)*PV
      DO 100 I=1,J
      IF (I-K) 90,100,95
   90 IJ = ITR2+I
      IK = ITR1 + I
      A(IJ) = A(IJ) - A(IK)*F
      GO TO 100
   95 IJ = ITR2 + I
      ITR3 = ITR3 + I - 1
      IK = ITR3 + K
      A(IJ) = A(IJ) - A(IK)*F
  100 CONTINUE
  110 CONTINUE
C
C     DIVIDE PIVOTAL ROW-COLUMN BY PIVOT, INCLUDING APPROPRIATE SIGNS
  120 ITR2 = ITR1
      DO 140 I=1,N
      IF (I-K) 125,130,135
  125 IK = ITR1+I
      A(IK) = -A(IK)*PV
      GO TO 140
C     (REPLACE PIVOT BY RECIPROCAL)
  130 A(KK) = PV
      GO TO 140
  135 ITR2 = ITR2+I-1
      KI = ITR2+K
      A(KI) =  A(KI)*PV
  140 CONTINUE
C
  145 CONTINUE
C
  150 RETURN
      END
```

REFERENCES

1. Sidik, Steven M.; and Henry, Bert: RAPIER-A FORTRAN IV Program for Multiple Linear Regression Analysis Providing Internally Evaluated Remodeling. NASA TN D-5656, 1970.

2. Wampler, Roy H.: A Report on the Accuracy of Some Widely Used Least Squares Computer Programs. J. Am. Stat. Assoc., vol. 65, no. 330, June 1970, pp. 549-565.

3. Golub, G.: Numerical Methods for Solving Linear Least Squares Problems. Numer. Math., vol. 7, 1965, pp. 206-216.

4. Hoerl, Arthur E.; and Kennard, Robert W.: Ridge Regression: Biased Estimation for Nonorthogonal Problems. Technometrics, vol. 12, no. 1, Feb. 1970, pp. 55-67.

5. Draper, N. R.; and Smith, H.: Applied Regression Analysis. John Wiley & Sons, Inc., 1966.

6. Kannenberg, Robert G.: CINEMATIC - FORTRAN Subprograms for Automatic Computer Microfilm Plotting. NASA TM X-1866, 1969.

7. Faddeeva, V. N.: Computational Methods of Linear Algebra. Dover Publications, 1959.

8. Kennedy, W. J.; and Bancroft, T. A.: Model Building for Prediction in Regression Based Upon Repeated Significance Tests. Ann. Math. Stat., vol. 42, no. 4, Aug. 1971, pp. 1273-1284.

9. Davies, Owen L.; ed.: The Design and Analysis of Industrial Experiments. Second ed., Hafner Pub. Co., 1956.

10. Box, G. E. P.; and Hunter, J. S.: Multi-factor Experimental Designs for Exploring Response Surfaces. Ann. Math. Stat., vol. 28, 1957, pp. 195-241.

11. Sandrock, Gary D.; and Holms, Arthur G.: Statistical Design and Analysis of Optimum Seeking Experiments to Develop a Gamma-Prime Strengthened Cobalt-Nickel Base Alloy. NASA TN D-5587, 1969.

12. Anon.: System/360 Scientific Subroutine Package (360A-CM-03X) Version II Programmer's Manual. Technical Publication Dept., IBM Corp., White Plains, N.Y., 1967.

13. Webb, S. R.; and Galley, S. W.: Design, Testing and Estimation in Complex Experimentation. Part IV: A Computer Routine for Evaluating Incomplete Factorial Designs. Rocketdyne Div., North American Aviation (ARL-65-116, Pt. IV, DDC No. AD-618518), June 1965.

NASA-Langley, 1972 —— 8 E-6718

NATIONAL AERONAUTICS AND SPACE ADMISTRATION
WASHINGTON, D.C. 20546

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE $300

FIRST CLASS MAIL

POSTAGE AND FEES PAID
NATIONAL AERONAUTICS AND
SPACE ADMINISTRATION

U.S.MAIL

005 001 C1 U 08 720421 S00903DS
DEPT OF THE AIR FORCE
AF WEAPONS LAB (AFSC)
TECH LIBRARY/WLOL/
ATTN: E LOU BOWMAN, CHIEF
KIRTLAND AFB NM 87117

POSTMASTER: If Undeliverable (Section 15. Postal Manual) Do Not Retu

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

— NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546